



DETERMINATION OF STATIC  
AND DYNAMIC STABILITY COEFFICIENTS  
USING BEGGAR

THESIS

Michael E. Bartowitz, Second Lieutenant, USAF

AFIT/GAE/ENY/08-M02

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY

**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GAE/ENY/08-M02

DETERMINATION OF STATIC  
AND DYNAMIC STABILITY COEFFICIENTS  
USING BEGGAR

THESIS

Presented to the Faculty  
Department of Aeronautics and Astronautics  
Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
In Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Aeronautical Engineering

Michael E. Bartowitz, B.S.A.E.  
Second Lieutenant, USAF

March 2008

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

DETERMINATION OF STATIC  
AND DYNAMIC STABILITY COEFFICIENTS  
USING BEGGAR

Michael E. Bartowitz, B.S.A.E.  
Second Lieutenant, USAF

Approved:



Lt Col R.C. Maple (Chairman)

12 Mar 08

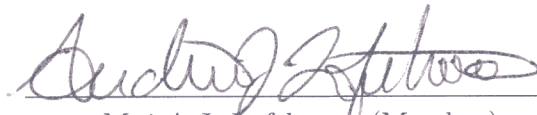
date



Lt Col C.M. Shearer (Member)

10 Mar 08

date



Maj A.J. Lofthouse (Member)

12 Mar 08

date

*Abstract*

The static and dynamic pitch and roll stability derivatives of a finned, axisymmetric missile known as the Basic Finner were examined using a Computational Fluid Dynamics (CFD) approach. Stability derivatives are used to characterize vehicle motion, and knowledge of them is critical to the design of stable uncontrolled vehicles and control systems for controlled vehicles. Using CFD to characterize the motion of new munition designs has the potential to improve overall performance and reduce research and testing costs. The present analysis simulated forced oscillation and free oscillation of the Basic Finner model using the Air Force SEEK EAGLE Office's Beggar code. The pitch stability derivatives were determined at  $0^\circ$  angle of attack for six Mach numbers from 1.58 to 2.50 and at Mach number equal to 1.96 for angles of attack from  $0^\circ$  to  $20^\circ$ . The parameters defining the motion of the forced oscillation tests were the reduced pitch rate, amplitude, Newton iterations, iterations per oscillation, and total oscillations. Convergence studies on each of these parameters were performed to ensure both convergence and solution independence. Roll stability derivatives were determined through forced, constant rate rolling motion for six Mach numbers from 1.58 to 2.50 at an angle of attack of  $0^\circ$ . The parameters defining the roll motion were reduced roll rate and iterations per revolution, which were chosen in the same manner as the pitch parameters. Good agreement was found between the different methods tested, previous CFD analysis, and experimental data.

## *Acknowledgements*

I would like to thank the denizens of the Linux lab: Abram Claycomb, Darrell Crowe, Jason Lee, Aaron McClung, and Nathan Pitcher. Without their help, I couldn't have both finished my thesis and remained sane. I would also like to express my thanks to my advisor, Lt Col Raymond Maple, for his valuable assistance and to Magdi Rizk and Lt Judson Babcock for their help on my many questions about Beggar. To my friends and family, thank you for your help and understanding during this difficult time. Finally, to my fiancée, I can't imagine accomplishing this, or anything else, without you.

Michael E. Bartowitz

# *Table of Contents*

	Page
Abstract . . . . .	iv
Acknowledgements . . . . .	v
List of Figures . . . . .	viii
List of Tables . . . . .	xi
List of Symbols . . . . .	xii
List of Abbreviations . . . . .	xiv
 I. Introduction . . . . .	 1
1.1 Research Goals . . . . .	3
1.2 Stability . . . . .	3
1.2.1 Static Stability . . . . .	4
1.2.2 Dynamic Stability . . . . .	5
1.3 Prior Research . . . . .	7
1.3.1 Linear Theory and Semi-Empirical Methods . . . . .	7
1.3.2 Experimental Methods . . . . .	9
1.3.3 CFD Approaches . . . . .	13
1.4 Research Approach . . . . .	15
1.5 Roadmap . . . . .	16
 II. Theory . . . . .	 17
2.1 Equations of Motion . . . . .	18
2.1.1 Aerodynamic Derivatives . . . . .	21
2.1.2 Pure Pitching Motion . . . . .	25
2.1.3 Pure Rolling Motion . . . . .	30
2.2 Governing Equation . . . . .	32
2.3 Solver Methods . . . . .	35
2.3.1 Boundary Conditions . . . . .	38
2.3.2 Overset Grids . . . . .	38
2.3.3 Six Degree of Freedom Model . . . . .	39

	Page
III. Methodology . . . . .	42
3.1 Basic Finner Model and Geometry . . . . .	42
3.1.1 Model Geometry . . . . .	42
3.1.2 Grid Generation . . . . .	43
3.2 Static Solutions . . . . .	48
3.3 Dynamic Pitch . . . . .	50
3.4 Dynamic Roll . . . . .	54
3.5 Beggar Inputs . . . . .	55
3.6 Stability Derivative Calculation . . . . .	56
3.6.1 Static Stability Derivatives . . . . .	56
3.6.2 Dynamic Stability Derivatives . . . . .	58
IV. Results and Discussion . . . . .	60
4.1 Static Solutions . . . . .	60
4.1.1 Inviscid . . . . .	60
4.1.2 Viscous . . . . .	65
4.2 Prescribed Motion Parameter Selection . . . . .	69
4.2.1 Pitch: Forced Oscillation . . . . .	69
4.2.2 Roll: Constant Revolution . . . . .	78
4.2.3 Test Parameter Recap . . . . .	80
4.3 Pitch Stability Derivatives: $\alpha = 0 - 20^\circ$ . . . . .	81
4.3.1 Static Pitch Stability . . . . .	82
4.3.2 Pitch Damping . . . . .	83
4.3.3 Trajectory Prediction . . . . .	84
4.4 Pitch Stability Derivatives: $M = 1.58 - 2.50$ . . . . .	87
4.4.1 Static Pitch Stability . . . . .	88
4.4.2 Pitch Damping . . . . .	88
4.5 Roll Damping Derivative . . . . .	89
V. Conclusions . . . . .	92
5.1 Future Research . . . . .	93
Appendix A. Listings . . . . .	95
A.1 Beggar Inputs . . . . .	95
A.2 Post-processing Tools . . . . .	100
Bibliography . . . . .	104



## *List of Figures*

Figure		Page
1.1.	Statically Stable, Unstable, and Neutral Equilibrium Positions	4
1.2.	Spring and Mass System . . . . .	5
1.3.	Examples of Stable and Unstable Dynamic Motion . . . . .	6
1.4.	Coning Motion . . . . .	11
2.1.	Sample pitch moment coefficient slopes for stable and unstable aircraft. . . . .	17
2.2.	Body-Fixed Axes . . . . .	19
2.3.	Induced moment due to change in angle of attack. . . . .	22
2.4.	Induced velocity due to pitching motion. . . . .	23
2.5.	Induced velocity due to rolling motion. . . . .	25
2.6.	Effect of pitch damping on pure pitching response. . . . .	30
2.7.	Beggar communication types between grids . . . . .	39
2.8.	Comparison of conventional CFD and controls axes. . . . .	40
3.1.	Basic Finner configuration (dimensions in calibers) . . . . .	43
3.2.	Three-dimensional Basic Finner model. . . . .	43
3.3.	Inviscid body grid domain. . . . .	45
3.4.	Inviscid fin grid superblock . . . . .	46
3.5.	Basic finner block assembly . . . . .	47
3.6.	Viscous body superblock . . . . .	48
3.7.	Sample residual plot: $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	50
3.8.	Sample moment coefficient history: $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	57
3.9.	Free oscillation trajectory and method for determining pitch damping sum. . . . .	58
4.1.	Inviscid residual convergence. . . . .	60
4.2.	Inviscid force and moment coefficient histories. . . . .	61
4.3.	Filled Mach contours, constant z-plane of symmetry. . . . .	62

Figure		Page
4.4.	Static pressure on the missile surface and a constant x-plane in the wake. . . . .	63
4.5.	Zebra plots of Mach number for the constant z-plane of symmetry.	64
4.6.	Inviscid pitching moment coefficient. . . . .	64
4.7.	Viscous residual convergence. $M = 1.96$ , $\alpha = 10^\circ$ . . . . .	65
4.8.	Viscous force and moment histories. $M = 1.96$ , $\alpha = 10^\circ$ . . . .	66
4.9.	Comparison of surface velocity and Mach contours for viscous and inviscid cases. $M = 1.96$ , $\alpha = 20^\circ$ . . . . .	67
4.10.	Comparison of surface flow and stagnation pressure for viscous and inviscid cases. $M = 1.96$ , $\alpha = 20^\circ$ . . . . .	68
4.11.	Viscous and inviscid pitching moment coefficient. . . . .	69
4.12.	$C_m$ vs $\alpha$ histories with varying pitch rate. $M = 1.96$ , $\alpha = 5^\circ$ . .	71
4.13.	Angle of attack, lift coefficient, and pitching moment coefficient histories. $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	72
4.14.	$C_m$ vs $k_q$ . . . . .	73
4.15.	Pitch stability coefficients as a function of $k_p$ for two values of iterations per oscillation. $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	74
4.16.	$C_m$ vs $\alpha$ for multiple amplitudes. $M = 1.96$ , $\alpha_o = 5^\circ$ . . . . .	75
4.17.	Pitch stability coefficients as a function of amplitude for two values of iterations per oscillation. $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	76
4.18.	Effect of Newton iterations on pitch stability coefficients. $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	77
4.19.	Multiple cycles show oscillation convergence. $M = 1.96$ , $\alpha = 5^\circ$	78
4.20.	Roll moment coefficient for differing rates. $M = 1.96$ , $\alpha = 5^\circ$ .	79
4.21.	Roll damping coefficient as a function of rate and number of iterations. $M = 1.96$ , $\alpha = 5^\circ$ . . . . .	79
4.22.	Pitching moment cycle for two angles of attack. $k_q = 2.5e - 4$ .	81
4.23.	Comparison of static stability coefficients, $M = 1.96$ . . . . .	82
4.24.	Comparison of dynamic stability coefficients, $M = 1.96$ . . . . .	84
4.25.	Comparison of oscillation trajectories. $M = 1.96$ , $\alpha_{start} = 20^\circ$ .	85

Figure		Page
4.26.	Comparison of oscillation trajectories. $M = 1.96$ , $\alpha_{start} = 5^\circ$ . .	86
4.27.	Static and dynamic stability for two Mach numbers. . . . .	87
4.28.	$C_m$ vs $\alpha$ cycles for multiple Mach numbers. . . . .	87
4.29.	Static pitch stability as a function of Mach number. . . . .	88
4.30.	Pitch damping as a function of Mach number. . . . .	89
4.31.	Roll moment convergence for six Mach numbers, $k_p = 0.0025$ . .	90
4.32.	Roll damping coefficient as a function of Mach number, $k_p =$ 0.0025. . . . .	91

# *List of Tables*

Table		Page
1.1.	Army/Navy Aerodynamic Prediction Code . . . . .	8
3.1.	Inviscid grid dimensions . . . . .	47
3.2.	Viscous grid dimensions . . . . .	49
3.3.	Flow and model parameters for static cases. . . . .	49
3.4.	Dynamic test cases . . . . .	54
4.1.	Effect of reduced pitch rate on stability coefficients: $1600 \frac{\text{iterations}}{\text{oscillation}}$ .	73
4.2.	Effect of iterations per revolution on roll moment and damping, $k_p = 0.0025$ . . . . .	80

# *List of Symbols*

Symbol		Page
$M$	Mach Number . . . . .	9
$Re$	Reynolds Number . . . . .	10
$V$	Velocity . . . . .	10
$a$	Speed of Sound . . . . .	10
$\rho$	Density . . . . .	10
$\mu$	Dynamic Viscosity . . . . .	10
$d$	Diameter . . . . .	10
$(\theta, \phi, \psi)$	Euler Angles: pitch, roll, yaw . . . . .	12
$C_m$	Pitching Moment Coefficient . . . . .	17
$\alpha$	Angle of Attack . . . . .	17
$C_l$	Roll Moment Coefficient . . . . .	18
$\mathbf{F}$	Force Vector . . . . .	18
$m$	Mass . . . . .	18
$\mathbf{M}$	Moment Vector . . . . .	18
$\mathbf{H}$	Angular Momentum Vector . . . . .	18
$p$	Roll Rate . . . . .	20
$q$	Pitch Rate . . . . .	20
$C_{m_\alpha}$	Static Pitch Stability Coefficient . . . . .	21
$Q = \frac{1}{2}\rho V^2$	Dynamic Pressure . . . . .	21
$S$	Reference Area . . . . .	21
$C_{m_q}$	Moment Coefficient due to Pitch Rate . . . . .	22
$C_{m_{\dot{\alpha}}}$	Moment Coefficient due to rate of change of angle of attack . . . . .	22
$k_q = \frac{qd}{2V_\infty}$	Reduced Pitch Rate . . . . .	23
$C_{l_p}$	Roll Damping Coefficient . . . . .	24
$I_y$	Moment of Inertia about the y-axis . . . . .	25

Symbol		Page
$\ddot{\theta}$	Angular Acceleration in Pitch . . . . .	25
$\dot{\alpha}$	Time Rate of Change of Angle of Attack . . . . .	26
$\dot{\phi} = q$	Pitch Rate . . . . .	26
$\delta_e$	Elevator Angle . . . . .	26
$M_q + M_{\dot{\alpha}}$	Pitch Damping Derivative Sum . . . . .	26
$M_{\alpha}$	Static Stability Derivative . . . . .	26
$\omega$	Damped Natural Frequency . . . . .	28
$\zeta$	Damping Ratio . . . . .	29
$\omega_n$	Undamped Natural Frequency . . . . .	29
$L_p$	Roll Damping Derivative . . . . .	31
$\vec{Q}$	Vector of Conserved Variables . . . . .	33
$\vec{F}_c$	Vector of Convective Fluxes . . . . .	33
$\vec{F}_v$	Vector of Viscous Fluxes . . . . .	33
$V_{\infty}$	Freestream Velocity . . . . .	51
$k_p$	Reduced Roll Rate, $\frac{pd}{2V_{\infty}}$ . . . . .	54

*List of Abbreviations*

Abbreviation		Page
CFD	Computational Fluid Dynamics . . . . .	1
(6+)DOF	Six Degree of Freedom . . . . .	3
AEDC	Arnold Engineering Development Center . . . . .	10
RBD	Rigid Body Dynamics . . . . .	14
N-S	Navier-Stokes . . . . .	32
ANSR	Army-Navy Spinner Rocket . . . . .	93

DETERMINATION OF STATIC  
AND DYNAMIC STABILITY COEFFICIENTS  
USING BEGGAR

## I. Introduction

Accurate performance estimates are critical to the efficient design of all engineering systems. It is important to assess the performance of new designs as early as possible in the design process in order to save time, money, and other resources. The design of aircraft and airborne weapons are no exception to this rule, and it is important to verify each new design as quickly, inexpensively, and safely as possible.

Performance analyses of missiles and other projectiles focus primarily on three areas: the launch system, the ability to accurately strike a target, and the amount of energy or damage delivered to the target. The present research will focus on a projectile's performance in the air and its ability to hit the desired target, also known as exterior ballistics [38]. In order to determine the aerodynamic performance of a new design, missile and projectile designers typically turn to flight tests, but as modern designs become more complicated, the testing process is becoming more complicated as well. Problems with complex geometries or extreme flow conditions can be very expensive, or even impossible, to test in a wind tunnel. The physical limitations of wind tunnels, combined with rising costs, may make wind tunnels insufficient for meeting the needs of future designs [12].

Computational fluid dynamics (CFD) eliminates all of the physical limitations and many of the other limitations associated with wind tunnel testing and has the potential to positively affect the cost, schedule, and safety of the design and validation of new flight systems. In some cases, CFD has been shown to effectively reduce the time and cost required to obtain aerodynamic data by as much as one year and



hundreds of thousands of dollars when compared to obtaining the same data using wind tunnels [12].

CFD is routinely used to resolve the static aerodynamics and flow characteristics of complex geometries. In the past, it was often considered sufficient to determine the static stability characteristics of objects in flight, and the dynamic stability coefficients were either assumed to be negligible, or they were treated as some small constant determined from simple approximations [22]. As modern designs become more complicated and the flight conditions experienced become more extreme, the dynamic stability characteristics are becoming increasingly important. This is particularly true on slender vehicles with fins, for which the damping derivatives have a strong influence on vehicle response at high speeds and high angles of attack [33].

Predicting dynamic stability derivatives has been a challenge since the Apollo and Viking programs [17]. Experimental methods have been available to determine dynamic stability coefficients, but these methods are expensive, and the resources needed to carry them out are very limited [22, 33]. The use of CFD as a tool for examining dynamic stability characteristics was somewhat limited in the past, because methods had not been proven or were deemed too computationally expensive. Recently, capabilities for predicting pitch and roll damping and Magnus moment coefficients have been developed, making it possible for both static and dynamic stability analysis to be performed using only CFD [34].

The use of CFD for determining the aerodynamic characteristics of missiles and projectiles is becoming more widespread. In the past, the unsteady flows and moving geometries associated with determining dynamic aerodynamic stability parameters made dynamic solutions more difficult to compute and less reliable, but modern methods and resources are making this process increasingly reasonable. Flight tests remain an essential ingredient for determining the aerodynamics of new designs, but the process of flight testing is both expensive and time-consuming, and it often cannot be completed early enough in the design process to have a sufficient impact. Modern

CFD methods and resources are both fast and accurate enough to greatly reduce the design costs and provide a detailed understanding of complex aerodynamics [28].

### **1.1 Research Goals**

The goal of this research is to test and verify the capabilities of the Beggar code for determining the static and dynamic stability coefficients of objects in flight. Specifically, the static pitch and roll stability coefficients and the dynamic pitch and roll damping moment coefficients will be determined for the Basic Finner missile model. Beggar is routinely used to calculate static stability coefficients or to run real time, coupled, six degree of freedom ((6+)DOF) store separations, and the code has the capabilities necessary to obtain the data used to determine the dynamic stability coefficients, but it is not customarily used to do so [12, 14, 19, 25, 26]. The present effort will expand the application of the Beggar code using its current capabilities.

### **1.2 Stability**

For an object in flight, the word “stability” refers to the tendency of that object to return to its equilibrium position after it has been disturbed. Some disturbances are intentional and are input by a pilot or a computer. Other disturbances are caused by atmospheric effects like wind gusts/gradients or turbulent air. Regardless of the cause of the disturbance, a missile or airplane that is stable in flight will return to its equilibrium position.

The equilibrium position is typically referred to as the trimmed condition for an aircraft. In order to achieve this condition, the sum of both the forces and the moments about the center of gravity must be zero. Once the equilibrium flight condition is reached, it remains unchanged unless acted upon by an outside force, such as the disturbances mentioned above. A statically stable system will respond to any disturbance with a force/moment that tends to move it back toward the equilibrium. If that system is also dynamically stable, then the equilibrium will eventually be reacquired. Otherwise, the system will diverge from its equilibrium position despite the

restoring force/moment. The concepts of static and dynamic stability are discussed further below.

*1.2.1 Static Stability.* Static stability is the tendency of an object to move back toward its equilibrium position after a disturbance. Figure 1.1 shows the classic example of static stability. When the ball is on flat ground, it is considered to have neutral static stability, because any point to which it is moved to will become a new equilibrium from which it will not stir unless disturbed. The ball on top of the hill is in an equilibrium position that is statically unstable. From this position, even the slightest disturbance will cause the ball to continue to roll down the hill. Finally, the lowest ball is in a statically stable equilibrium position. Whichever way the ball is moved, the force of gravity will cause it to move back toward the original equilibrium position at the bottom of the hill.

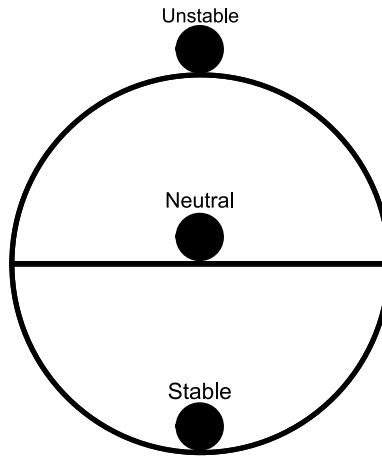


Figure 1.1: Statically Stable, Unstable, and Neutral Equilibrium Positions

Figure 1.2 shows a spring-mass system that will be familiar to most engineers. The equation of motion for this system is:

$$m\ddot{x} + kx = 0 \quad (1.1)$$

where  $m$  is the mass and  $k$ , the spring constant, is the static stability coefficient. In this case, a spring constant of zero causes the system to be neutrally stable: at

equilibrium regardless of the displacement. If  $k$  is less than zero, the force of the spring will reinforce any disturbance, and the mass will diverge from its equilibrium position. If  $k$  is greater than zero, the spring will act as a restoring force, moving the mass back toward the original equilibrium [13].

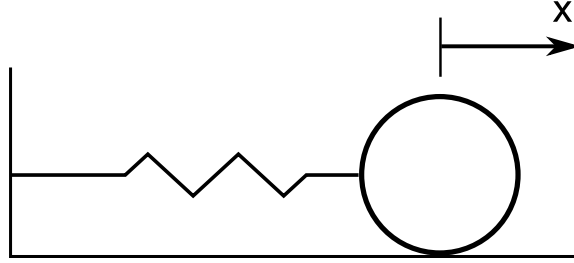


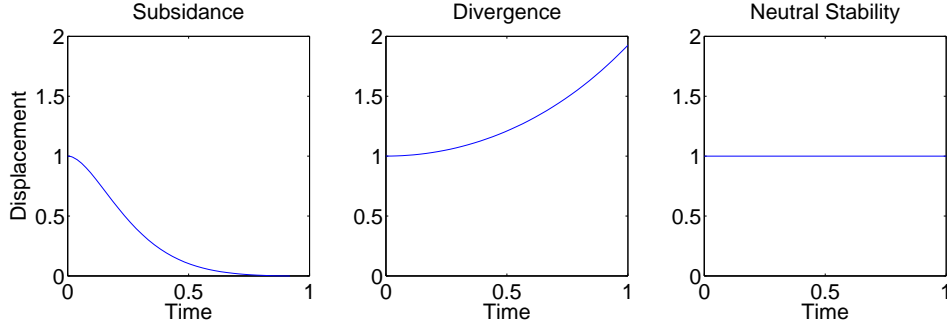
Figure 1.2: Spring and Mass System

*1.2.2 Dynamic Stability.* Static stability does not guarantee dynamic stability. Dynamic stability deals with the time history of a system after a disturbance. In order to be considered dynamically stable, the system must eventually return to the original equilibrium condition. Despite the restoring force of static stability, a return to the original equilibrium is not guaranteed, because static stability says nothing about whether the motion will ever settle out. It is possible for the spring-mass system in Figure 1.2 to oscillate indefinitely about the equilibrium position if there is an initial displacement. We know that this is not likely to happen in real life, however, because in a real system, some energy is typically removed from the motion of the system—an effect known as damping. With damping proportional to the velocity included, Equation 1.1 becomes

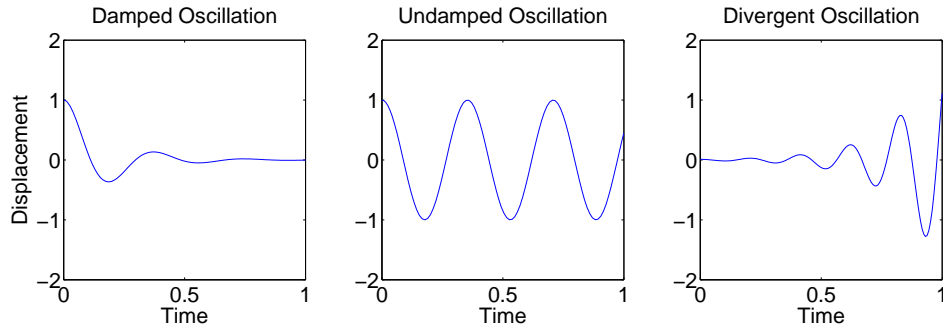
$$m\ddot{x} + b\dot{x} + kx = 0, \quad (1.2)$$

where  $b$  is the damping coefficient [13]. When  $b$  is greater than zero, the damping opposes the motion and energy is removed from the system. If  $k$  is also greater than zero, the system will be both statically and dynamically stable, and it will eventually return to its equilibrium position after being disturbed. The left-most images of

Figures 1.3(a) and (b) show examples of cases with positive values for both damping and stiffness. The non-oscillatory case has a damping coefficient much greater than the spring constant and is overdamped. The oscillatory case has  $k > b$ , causing damped oscillation.



(a) Non-Oscillatory Motion



(b) Oscillatory Motion

Figure 1.3: Examples of Stable and Unstable Dynamic Motion

If the spring constant is less than zero, the system will diverge from its equilibrium position, regardless of the damping coefficient, as shown in the middle image of Figure 1.3(a). An example of such a situation is the linear approximation of an inverted pendulum in a viscous fluid. Once the pendulum is disturbed from its statically unstable equilibrium at the top, it will continue to move away from the equilibrium, even though the viscous fluid opposes the motion [13].

If the damping coefficient is equal to zero, the response will be an undamped oscillation. For damping coefficients less than zero, energy is added to the system

as it moves, and it will be dynamically unstable, regardless of the value of  $k$ . The right-most image of Figure 1.3(b) shows an example of a system that is statically stable, but its negative damping coefficient causes it to be dynamically unstable [18].

An example of a case that is statically stable but dynamically unstable and behaves as though the damping were negative is a wing in flutter condition. The stiffness of the wing tends to bring the wing back toward its equilibrium position, but the unsteady flow field producing the flutter can cause the motion of the wing to diverge and fail catastrophically.

### **1.3 Prior Research**

There are three general methods that may be used to compute dynamic stability derivatives: approximations from linear theory or semi-empirical methods, ballistic or wind tunnel testing, and, more recently, CFD. Methods using linear theory or empirical data are typically the fastest and easiest, but there is often error associated with these methods, especially for complicated geometries or extreme flow conditions. Ballistic and wind tunnel testing can effectively provide accurate values, but finding those values can be both expensive and time consuming. CFD can provide the most effective means of calculating these derivatives because of its flexibility, speed, and accuracy.

*1.3.1 Linear Theory and Semi-Empirical Methods.* Missile designers have long recognized the necessity of evaluating the aerodynamic characteristics of a new design when it is still in the preliminary or conceptual design phase. It is sometimes difficult to build a scale model or even know exact geometries in the earliest phases, however, so rapid, inexpensive, easy to use methods for estimating important aerodynamic parameters are desired. Out of this desire arose codes like the Aerodynamic Prediction Code, developed in 1971 by the Army and Navy [9]. This code was designed to handle basic wing-body-tail configurations, which covered a large percentage of the tactical weapons in use at the time.

The Aerodynamic Prediction Code used a combination of many theoretical approximations and empirical data to attain its estimations. Table 1.1 shows these methods, along with the cases and applicable components. The code calculated aerodynamic force and moment components on each of the system components separately, and then added the contributions together while attempting to account for interference effects.

Table 1.1: Methods used by Army/Navy Aerodynamic Prediction Code for computing dynamic derivatives [9].

Component	Mach Number Region			
	Subsonic	Transonic	Low Supersonic	High Supersonic
<b>Body-Alone Pitch Damping Moment</b>	Empirical or Modified Slender-Body Theory	Empirical or Modified Slender-Body Theory or Linear Interpolation	Empirical or Embedded Newtonian Theory or Linear Interpolation	Empirical or Embedded Newtonian Theory
<b>Wing and Interference Roll Damping</b>	Lifting Surface Theory	Empirical	Linear Thin-Wing Theory	Strip Theory
<b>Body-Alone Magnus Moment</b>	Empirical			
<b>Wing and Interference Magnus Moment</b>	Assumed Zero			
<b>Body-Alone Roll Damping Moment</b>	Empirical			
<b>Wing and Interference Pitch Damping Moment</b>	Slender-Wing or Lifting Surface Theory	Slender-Wing or Supersonic Slender-Wing Theory or Empirical	Supersonic Slender-Wing or Linear Thin-Wing or Embedded Newtonian Strip Theory	Strip or Embedded Newtonian Strip Theory

These methods worked well for basic geometries and angles of attack, providing fast, reliable results that were used to guide new designs in order to optimize configurations, control gains, and performance. The major problem with this code, and others like it, lies in the fact that it is limited to very general conditions. This particular code was initially limited to Mach numbers less than three and angles of

attack below  $15^\circ$ . Those restrictions were later expanded [9], but strict geometric restrictions remained in place.

The breakdown in theory for more complex geometries, as well as the lack of empirical data for completely new designs, makes it impossible for codes like the Aerodynamic Prediction Code to attain accurate predictions for anything but the most basic geometries. To deal with complex geometries, more flexible methods are desired.

*1.3.2 Experimental Methods.* Ground based experiments and flight testing remain the most commonly utilized and trusted methods for obtaining the stability derivatives of objects in flight. In theory, perfect flight testing has the ability to exactly match the flight conditions of a missile or projectile, and can thus be used to determine exact responses with no approximations. In reality, the mission flight conditions can be difficult to simulate, and responses are often very difficult to measure accurately.

*1.3.2.1 Ground Based Tests.* Historically, two types of ground based testing have been used to determine a model's aerodynamic stability characteristics. These methods are ballistic range testing and wind tunnel testing. Both methods have the advantage of using actual models in actual flows; this lends credibility to the tests, because any simulation can only approximate real situations. This is also a disadvantage, however, because good models are often very expensive and time-consuming to fabricate. Depending on the type of testing employed, other challenges and limitations also exist. Some of these issues are discussed below.

In either case, the models tested are typically scaled down in size, which often changes the inertial properties from those of the actual missile or projectile. Because of this scaling, it is necessary to match certain flow properties in order to duplicate full scale flight conditions or to compare data from different facilities. The most important of these are the non-dimensional numbers known as Mach Number ( $M$ ) and



Reynolds Number ( $Re$ ), where  $M = V/a$  is the velocity ( $V$ ) non-dimensionalized by the speed of sound ( $a$ ), and  $Re_L = \rho V L / \mu$  is the ratio of inertial forces to viscous forces where  $\rho$  is the density,  $L$  is the reference length, and  $\mu$  is the dynamic viscosity [38]. When dealing with missiles and projectiles, the diameter,  $d$ , is commonly used as the reference length for the Reynolds Number. All Reynolds numbers in the present work use this convention.

In wind tunnel testing, rather than move the model of interest through the air, the air is moved over the model. This is a valid approach because, in general, the interaction between the model and the flow is independent of which is moving. This is true in most cases, but the quality of the flow in a wind tunnel can still be affected by many things. Some of the issues involved in wind tunnel testing include wall effects, non-uniform flow, turbulence, and sting effects caused by the mounting system [7,38]. These effects can be accounted for in some ways, like turbulence tripping and flow conditioning, but results may still be negatively affected.

Resolving forces and moments in wind tunnels requires precise six component force and moment balances. The equipment used to take these measurements can be quite complicated and expensive to install and use, especially for the dynamic tests required to capture the dynamic stability coefficients. In wind tunnels, there are three types of dynamic tests that may be used to determine the dynamic pitch stability coefficients: planar forced oscillation, planar free oscillation, and steady-state coning motion. Forced oscillation does not allow the model to respond to the flow of the air, but measures the forces and moments on the model while maintaining a constant frequency of oscillation. Accomplishing this requires models with very low moments of inertia, and only a few facilities are capable of testing in this manner [7].

Free oscillation starts the model at some angle of attack and then allows it to oscillate freely about a trim condition. Uselton [33] performed small amplitude free oscillation tests on the Basic Finner model at the Arnold Engineering Development Center's (AEDC) von Kármán Dynamics Facility, which was equipped with a spe-

cialized test mechanism that used a strut-supported cross-flexure balance in order to measure the dynamic forces and moments. This data was used to validate the methods used in the present research. When using free oscillation, it is critical that the model moments of inertia closely match the real moments of inertia, because they can affect the free-flight angular motion [38].

In steady-state coning motion, the longitudinal body-fixed axis of the missile is rotated about a vector that is parallel to the flow and passes through the center of gravity at a constant angular rate,  $\Omega$ , as shown in Figure 1.4. The model may also rotate about its longitudinal body-fixed axis at an angular rate,  $\omega$ . For the special case when  $\omega = 0$ , the coning motion is referred to as lunar coning [8]. The model's stability coefficients may be calculated based on the side moment experienced and the angular rotation rate. Since the angular rate is constant and the solution is run to steady state, the coning motion method does not have to account for transients, which can eliminate one source of error [7]. Other complications exist, however, such as accurately measuring the forces and moments while rotating the model, sometimes about two axes. Facilities capable of this type of experiment are limited in number, but the method is growing in popularity.

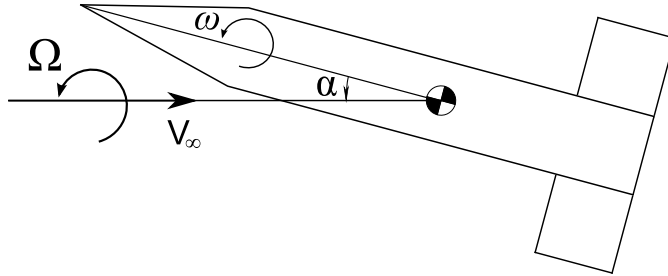


Figure 1.4: Coning Motion

Aeroballistic test ranges, also known as free-flight spark ranges, do not share all of the same issues as wind tunnels do, because the projectile is the moving component, going through a still atmosphere. To a large degree, this eliminates problems with wall effects, non-uniform flow, and interference from the mounting system. Free-flight spark ranges typically consist of a gun room, a blast chamber, and a long, sometimes

enclosed area called the range [38]. Projectiles are launched from the gun room and blast chamber, attempting to simulate the speeds and flight conditions that the projectile would experience when carrying out its mission. Once the blast chamber is cleared a projectile flies unconstrained through the range. Throughout the range, stations are set up orthogonal to the flight path that contain spark photography equipment and other measuring devices. The photographs from each of the spark photography stations are used to build a history of the projectile’s three dimensional position components,  $(\bar{x}, \bar{y}, \bar{z})$ , which define the location of the projectile’s center of gravity, and the three Euler angles,  $(\theta, \phi, \psi)$ , which define the projectile’s orientation [5].

The position and orientation data gathered at discrete locations in the range are smoothly interpolated between points to build a continuous time-history of the flight. Boissevain and Intrieri [5] used the history of the position and orientation elements to determine the angles of attack and side-slip, which, in turn, were used to solve assumed forms of the equations of motion. This method has since been updated and, to a large degree, automated to determine the coefficients of the equations of motion by fitting a curve to the position and orientation data [7, 38]. The CADRA2 interactive software, developed by Yates [7], has been used to provide these fits for data gathered in Eglin Air Force Base’s Aeroballistic Research Facility. The software accepts trajectory data as an input, allows the user to specify the assumed form of the equations of motion, and outputs the unknown coefficients for that assumed form.

These methods can be effective, but, like wind tunnels, the facilities capable of carrying out these tests are very limited, and they can be expensive and difficult to use. Another issue shared with wind tunnels is the effect of scaling. Sizes, weights, forces and moments can be effectively scaled, but boundary layers and turbulent effects do not scale. Because ballistic ranges and wind tunnels often use sub-scale models, it is possible for the control surfaces of the models tested in these facilities to be submersed in the boundary layer, and the measured effectiveness could be greatly impacted [38].

Ground-based test facilities attempt to match real conditions as much as possible, and, in general, they do an acceptable job determining the aerodynamics of new designs. The process can involve many challenges, however, and is difficult to accomplish fast enough to effect fundamental design changes. This fact, coupled with the fact that the tests are often quite expensive to run, makes an alternative desirable.

*1.3.2.2 Flight Tests.* Actual flight tests with full scale models may seem like the optimal method of determining stability characteristics, but flight testing can sometimes prove to be difficult. To begin with, live flight experiments are typically expensive, time-consuming, and may even be dangerous. Also, there are a number of factors that are difficult to account for in flight tests, such as wind gusts and changing atmospheric properties [7]. Even without those uncertain effects to deal with, data acquisition on real systems can be difficult because of relative velocities, since it is not possible to have relevant cameras or other sensors set up at fixed locations with respect to the flight path. This makes flight testing very difficult, but, when properly performed, flight testing provides the ultimate verification of a design's performance. Unfortunately, because of the difficulty, time, and expense involved in effective flight testing, it is generally not possible for flight tests to have an impact on the design phase when significant changes may still be made.

*1.3.3 CFD Approaches.* Because experimental approaches are time-consuming and expensive, and theoretical and empirical approaches are not flexible enough, designers have turned to CFD for determining the aerodynamic characteristics of missiles and projectiles. Physical testing remains a valuable tool for the validation of computational data and techniques, but CFD approaches may be applied to new designs of any shape or size relatively early in the conceptual design phase without the need for actual model fabrication.

The utility of CFD lies in the fact that it can be used to simulate exact conditions. Because of this, any type of testing that is possible in wind tunnels or test ranges may also be simulated using CFD, so long as an appropriate solver is used.

This allows CFD approaches to employ the same techniques for determining the static and dynamic stability coefficients as wind tunnels and ballistic ranges.

Ballistic trajectories are simulated by providing initial conditions and then using a real-time, six degree of freedom solver to build a continuous time-history of a projectile's flight. The time-history is constructed by coupling the forces and moments found in the CFD solver with a rigid body dynamics (RBD) model to solve for the motion [27]. This data may be processed to determine the stability coefficients with a tool like CADRA2 [7] just like the data from a ballistic range. The main difference is that the ballistic range captures the projectile's position and orientation at discrete points and interpolates to build the time-history, while the data from a CFD trajectory will be nearly continuous and requires no interpolation. Data found with this method has been found to agree very well with data from test ranges [27]. An issue with this method, however, is that a full (6+)DOF trajectory can be computationally expensive to compute, and multiple trajectories are often needed to accurately determine the stability coefficients.

A more common and less computationally expensive method for determining the dynamic stability coefficients is forced oscillation. Just like the method used in wind tunnels, forced oscillation involves pre-defined planar rotation about the model center of gravity. In various manners, CFD has been used to apply forced oscillation [8,16,17,21,28–30]. Sahu [27–30], DeSpirito, Silton, and Weinacht [8], and Oktay and Akay [21] have used forced oscillation to determine the damping coefficients on various missile models, both finned and unfinned. Both viscous and inviscid models were tested for various Mach numbers, typically determining the coefficients at an angle of attack of  $0^\circ$ . In general, good agreement was found both between similar computational tests and experimental approaches. Murman [16] applied small amplitude forced oscillations to various projectiles at varying angles of attack to examine the change in damping using a reduced-frequency approach. The results were found to match closely with experimental data.

Murman and Aftosmis [17] performed planar free oscillation testing on the Basic Finner model. The model was started at an angle of attack of  $20^\circ$  and a coupled CFD/RBD solver was used to predict the damped oscillations about the trim angle of  $0^\circ$  degrees. While real-time, coupled free oscillation tests take longer than a single forced oscillation test, the advantage of free oscillation lies in the fact that the damping at any angle of attack that the model passes through more than once may be determined from a single test. Forced oscillation testing requires that a small amplitude oscillation test be carried out at each angle of attack of interest. Murman and Aftosmis [17] found free and forced oscillation techniques to be in good agreement with each other and with experimental data, especially at low to moderate angles of attack.

Time-accurate approaches, while effective for determining damping coefficients, bring with them both computational expense and some degree of complication. The steady state coning method can determine the pitch damping and eliminates the need for time-accuracy. Weinacht [34] and DeSpirito et al [8] applied a combination of coning and lunar coning to various spin-stabilized projectiles. The steady state solutions were converged and the pitch damping predicted was found to compare very well to experimental data and to time-accurate methods, except at low supersonic Mach numbers, where the experimental data had a large degree of scatter.

#### ***1.4 Research Approach***

The current effort applied forced oscillation and free oscillation techniques to the Basic Finner model using the Beggar CFD code. The Basic Finner model and grids were built using Gridgen<sup>®</sup>. Static solutions were computed at various Mach numbers and angles of attack, and those solutions were used as the starting point of various dynamic tests. The results of the dynamic tests were used to obtain a history of the moment coefficients, which, in turn, were used to calculate the stability derivatives for each case. These results were compared to previously accomplished experimental and computational tests.

## ***1.5 Roadmap***

The flight equations of motion are presented in Chapter II, along with the derivation of the static and dynamic stability coefficients for pitch and roll. An overview of the Beggar solver is also given, along with a description of its methods for solving the governing equations, overset grid capability, and (6+)DOF model. Chapter III outlines the methods used in this research, from grid building to dynamic testing and stability coefficient extraction. Results are presented in Chapter IV, along with a discussion of the findings and comparison to previous findings from wind tunnels, ballistic ranges, and other CFD applications. Chapter V includes the conclusions reached from this testing and recommends possible courses of future research. The appendices present additional methods and results that are relevant to the research but not shown in Chapters III or IV.

## II. Theory

For an aircraft, just like any other system, static stability requires that the forces and moments generated in response to a disturbance tend to move the system back toward the equilibrium position. For an aircraft to be statically stable in pitch (known as longitudinal stability), this means that the aircraft's response to a nose up disturbance must be a nose down moment. Figure 2.1 shows example pitching moment coefficient ( $C_m$ ) curves versus angle of attack ( $\alpha$ ) for two aircraft that share an equilibrium at point B.

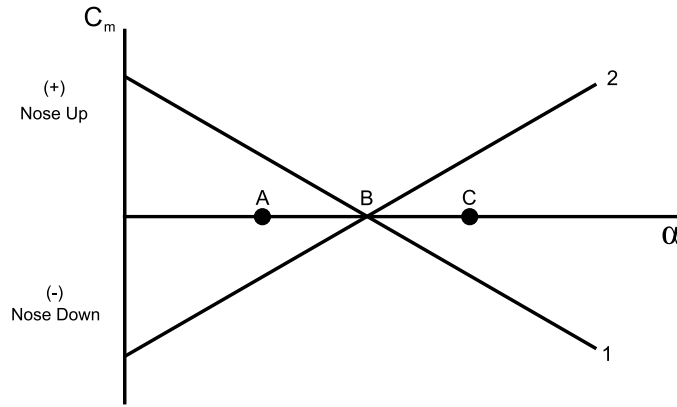


Figure 2.1: Sample pitch moment coefficient slopes for stable and unstable aircraft.

If a downward wind gust or another disturbance were to cause the angle of attack to move to point A, aircraft 1 would respond with a nose up pitching moment, moving back toward the original angle of attack. Aircraft 2, on the other hand, would respond to the decreased angle of attack with a nose down moment, further decreasing the angle of attack and causing the aircraft to diverge. If an upward gust changed the angle of attack to point C, the result would be the same, and aircraft 1 would again be forced back toward equilibrium, while the angle of attack of aircraft 2 would again diverge. This example demonstrates that the requirement for static pitch stability is [18]

$$\frac{dC_m}{d\alpha} < 0 \quad (2.1)$$



The requirements for static roll stability may be similarly discovered, resulting in a requirement that

$$\frac{dC_l}{d\phi} < 0 \quad (2.2)$$

where  $C_l$  is the roll moment coefficient, and  $\phi$  is the roll angle.

As discussed in the previous chapter, a system may be statically stable but never return to its original equilibrium position because of a dynamic instability. An airplane is considered dynamically stable only if the motion due to a disturbance decreases with time [18]. The degree of dynamic stability or instability depends upon the frequency of the oscillation and the speed with which motion either increases or decreases. In order to understand and predict static and dynamic stability, we must first understand the equations of motion governing aircraft flight.

## 2.1 *Equations of Motion*

The rigid body equations of motion are derived from Newton's 2<sup>nd</sup> Law, which can be expressed in vector form as:

$$\Sigma \mathbf{F} = \frac{d}{dt}(m\mathbf{v}) \quad (2.3)$$

$$\Sigma \mathbf{M} = \frac{d}{dt}\mathbf{H} \quad (2.4)$$

Simply put, the sum of the external forces,  $\mathbf{F}$ , is equal to the time rate of change of the linear momentum (the product of mass,  $m$ , and the velocity vector,  $\mathbf{v}$ ), and the sum of the external moments,  $\mathbf{M}$ , is equal to the time rate of change of the angular momentum,  $\mathbf{H}$ . These equations are frequently broken into scalar form with components along the body-fixed axes shown in Figure 2.2, yielding six separate equations. In the x, y, and z-directions, respectively, the components of each of the vectors above are:  $\mathbf{F} = [F_x \ F_y \ F_z]$ ,  $\mathbf{v} = [u \ v \ w]$ ,  $\mathbf{M} = [L \ M \ N]$ , and  $\mathbf{H} = [H_x \ H_y \ H_z]$ .

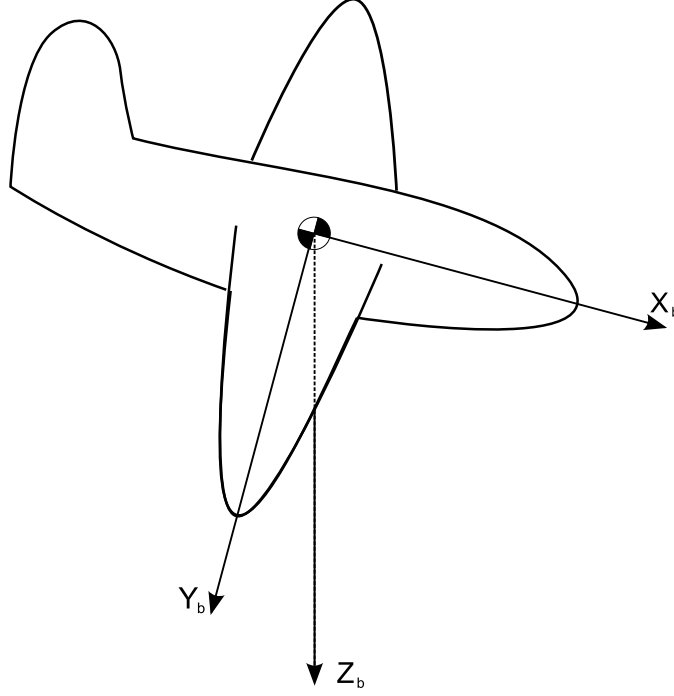


Figure 2.2: Body-Fixed Axes

If the mass of the vehicle is assumed to be constant and the angular velocity of the body frame is  $\omega$ , Equations 2.3 and 2.4 may be rewritten in terms of  $\omega$  and the velocity of the center of mass ( $\mathbf{v}_{\mathbf{cg}}$ ):

$$\mathbf{F} = m \frac{d\mathbf{v}_{\mathbf{cg}}}{dt} + m (\omega \times \mathbf{v}_{\mathbf{cg}}) \quad (2.5)$$

$$\mathbf{M} = m \frac{d\mathbf{H}}{dt} + \omega \times \mathbf{H} \quad (2.6)$$

In scalar form, assuming an axisymmetric projectile about the xy- and xz-planes, the equations are

$$\begin{aligned}
F_x &= m(\dot{u} + qw - rv) \\
F_y &= m(\dot{v} + ru - pw) \\
F_z &= m(\dot{w} + pv - qu)
\end{aligned} \tag{2.7}$$

$$\begin{aligned}
L &= I_x \dot{p} + qr(I_z - I_y) \\
M &= I_y \dot{q} + rp(I_x - I_z) \\
N &= I_z \dot{r} + pq(I_y - I_x)
\end{aligned} \tag{2.8}$$

where  $p$ ,  $q$ , and  $r$  are the scalar components of the angular velocity vector about the  $x$ ,  $y$ , and  $z$  axes, respectively, and  $I_x$ ,  $I_y$ , and  $I_z$  are the mass moments of inertia of the body about the axes. A full derivation of the flight equations of motion may be found in references [18] and [32].

By assuming that the motion of the projectile consists of small changes from a steady flight condition, small-disturbance theory may be applied to rewrite the equations of motion in terms of aerodynamic stability derivatives. In small-disturbance theory, each variable in the equation of motion is replaced by the sum of a reference value (denoted by a subscript '0') and a perturbation (denoted by a  $\Delta$ ) as, for example,  $u = u_0 + \Delta u$ . The aerodynamic stability derivatives are the incremental changes in forces or moments due to an incremental change from one of the reference conditions for velocity, acceleration, angular velocity, or control surface deflection. The small-disturbance theory equations of motion based on the aerodynamic derivatives that are most important for uncontrolled aircraft are:

$$\begin{aligned}
\Delta F_x &= \frac{\partial F_x}{\partial u} \Delta u + \frac{\partial F_x}{\partial w} \Delta w \\
\Delta F_y &= \frac{\partial F_y}{\partial v} \Delta v + \frac{\partial F_y}{\partial p} \Delta p + \frac{\partial F_y}{\partial r} \Delta r \\
\Delta F_z &= \frac{\partial F_z}{\partial u} \Delta u + \frac{\partial F_z}{\partial w} \Delta w + \frac{\partial F_z}{\partial \dot{w}} \Delta \dot{w} + \frac{\partial F_z}{\partial q} \Delta q
\end{aligned} \tag{2.9}$$

$$\begin{aligned}
\Delta L &= \frac{\partial L}{\partial v} \Delta v + \frac{\partial L}{\partial p} \Delta p + \frac{\partial L}{\partial r} \Delta r \\
\Delta M &= \frac{\partial M}{\partial u} \Delta u + \frac{\partial M}{\partial w} \Delta w + \frac{\partial M}{\partial \dot{w}} \Delta \dot{w} + \frac{\partial M}{\partial q} \Delta q \\
\Delta N &= \frac{\partial N}{\partial v} \Delta v + \frac{\partial N}{\partial p} \Delta p + \frac{\partial N}{\partial r} \Delta r
\end{aligned} \tag{2.10}$$

where the partial fractions are the aerodynamic stability derivatives.

*2.1.1 Aerodynamic Derivatives.* The stability derivatives of interest in the present analysis are those in the pitch and roll directions. The motion in the pitch direction is a function of both static and dynamic aerodynamic derivatives, but pure roll motion for an axisymmetric projectile at an angle of attack of zero degrees is governed only by a dynamic stability derivative.

*2.1.1.1 Pitch Derivatives.* The static pitch stability coefficient,  $C_{m_\alpha}$ , arises due to changing forces on the projectile as the angle of attack is changed.  $C_{m_\alpha}$  is also known as the pitch stiffness, and it acts like the spring constant term found in Equation 1.2. This term is defined as:

$$C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha} = \frac{\partial M / \partial \alpha}{Q S d / I_y} \tag{2.11}$$

where  $Q = \frac{1}{2} \rho V^2$  is known as the dynamic pressure,  $d$  is the reference length, the missile diameter, and  $S$  is the reference area, which is the area of the base for missiles, given as  $\frac{\pi d^2}{4}$ .

As discussed above, in order to be statically stable in pitch, the slope of the  $C_{m_\alpha}$  curve must be negative. This opposes any deviations and forces the aircraft back toward equilibrium when it is disturbed. For standard configuration missiles and aircraft, this negative moment is developed by a change in the lift on the tail. Figure 2.3 shows the change in lift on the tail, and thus the change in moment about the center of gravity, that results from a change in angle of attack.

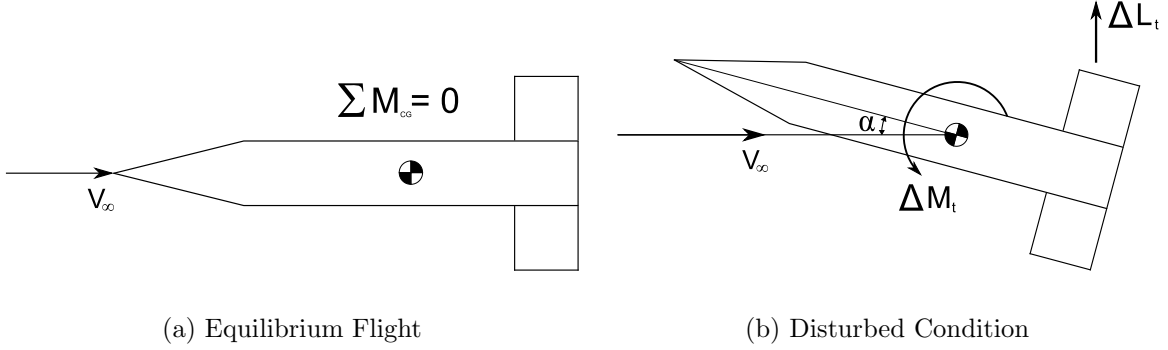


Figure 2.3: Induced moment due to change in angle of attack.

Two dynamic derivatives contribute to dynamic pitch stability. These are the moment coefficient due to pitch rate,  $C_{m_q}$ , and moment coefficient due to the rate of change of angle of attack,  $C_{m_{\dot{\alpha}}}$ . These terms are defined in terms of the angular rates  $q$  and  $\dot{\alpha}$  nondimensionalized by the diameter and freestream velocity, which are generally treated as constants [32].

$$C_{m_q} = \frac{\partial C_m}{\partial \left( \frac{qd}{2V_\infty} \right)} \quad \text{and} \quad C_{m_{\dot{\alpha}}} = \frac{\partial C_m}{\partial \left( \frac{\dot{\alpha}d}{2V_\infty} \right)} \quad (2.12)$$

The  $C_{m_q}$  stability coefficient develops in response to a change in the effective angle of attack on the tail due to the pitching rate. Figure 2.4 shows a missile model pitching up, which causes the tail to see an upward induced velocity equal to the product of the pitch rate and the distance from the tail to the center of gravity. For supersonic cases, the forward velocity,  $V_\infty$ , can generally be assumed to be much greater than the induced velocity, so the change in effective angle of attack is  $\Delta\alpha = \frac{ql_t}{V_\infty}$ , as shown in Figure 2.4.

$$\Delta\alpha = \frac{ql_t}{V_\infty} \quad (2.13)$$

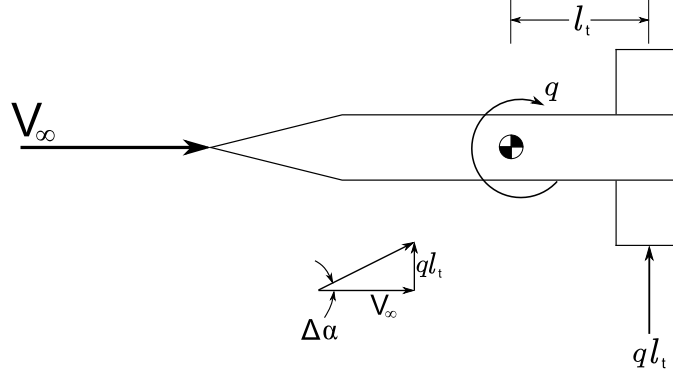


Figure 2.4: Induced velocity due to pitching motion.

This change in angle of attack results in a change in the lifting force on the tail as

$$\Delta L_t = C_{L\alpha_t} \frac{ql_t}{V_\infty} QS \quad (2.14)$$

The change in pitching moment due to the change in lift on the tail, then, is

$$\Delta M_{cg} = -l_t \Delta L_t \quad (2.15)$$

This shows that, for the example in Figure 2.4, pitching up motion develops a pitching down moment that opposes the motion. This pitching moment may be expressed in terms of a change in moment coefficient as

$$\Delta C_{m_{cg}} = \frac{\Delta M_{cg}}{QSd/I_y} = -\frac{C_{L\alpha} ql_t^2}{V_\infty d} \quad (2.16)$$

This relation may be expressed as an aerodynamic derivative due to a nondimensional pitch rate,  $k_q = \frac{qd}{2V_\infty}$ , known as the reduced pitch rate.

$$C_{m_q} \equiv \frac{\partial C_m}{\partial \left( \frac{qd}{2V_\infty} \right)} = \frac{2V_\infty}{d} \frac{\partial C_m}{\partial q} = \frac{2V_\infty}{d} \frac{\partial M / \partial q}{QSd/I_y} \quad (2.17)$$

The other dynamic pitch stability coefficient,  $C_{m_{\dot{\alpha}}}$ , is due to the time rate of change of angle of attack, which may be different than the pitch rate because of plunging or lag in the down-wash reaching the tail. For the conventional missile configurations considered here, there is no appreciable down-wash to affect the  $\dot{\alpha}$  term, so  $C_{m_{\dot{\alpha}}}$  may be determined similarly to  $C_{m_q}$

$$C_{m_{\dot{\alpha}}} \equiv \frac{\partial C_m}{\partial \left( \frac{\dot{\alpha}d}{2V_\infty} \right)} = \frac{2V_\infty}{d} \frac{\partial C_m}{\partial \dot{\alpha}} = \frac{2V_\infty}{d} \frac{\partial M / \partial \dot{\alpha}}{QSd/I_y} \quad (2.18)$$

In practice, it is very difficult to measure these two dynamic stability coefficients separately, so it is commonly considered sufficient to determine the sum of the two [33]. This sum,  $(C_{m_q} + C_{m_{\dot{\alpha}}})$ , is known as the pitch damping stability coefficient.

*2.1.1.2 Roll Derivatives.* The dynamic roll stability derivative of interest is the roll response due to roll rate,  $C_{l_p}$ . This term is known as the roll damping coefficient, and is defined as

$$C_{l_p} = \frac{\partial C_l}{\partial \left( \frac{pd}{2V_\infty} \right)} \quad (2.19)$$

Roll damping arises because of an uneven lift distribution created by the rolling motion. Figure 2.5 illustrates the linear velocity distribution developed by rolling motion, which, in turn, develops a linear distribution of local angle of attack and lift. The induced velocity is equal to the product of the distance from the center of gravity,  $y$ , and the roll rate,  $p$ . Note that the forces developed by the motion produce a roll moment that opposes the rolling motion and is proportional to the roll rate and the distance from the center of gravity. By inspection, we see that missiles with larger fins and faster spin rates experience more roll damping. Additional roll stability

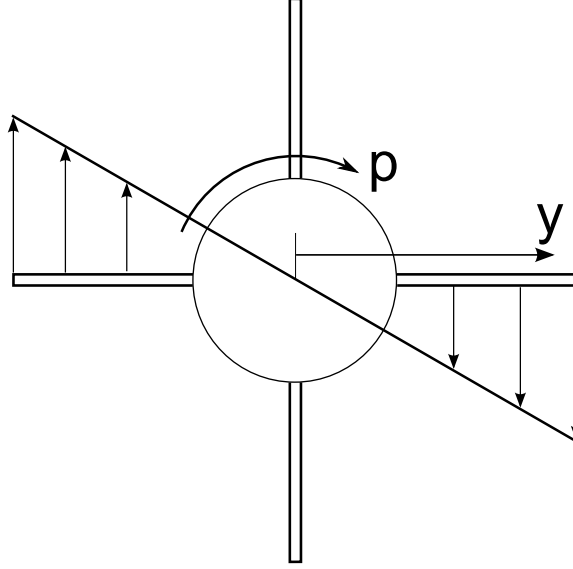


Figure 2.5: Induced velocity due to rolling motion.

derivatives develop due to sideslip and angle of attack, but the present analysis will be limited to pure rolling motion.

*2.1.2 Pure Pitching Motion.* For a special case in which a missile is constrained to fly in a straight line at a constant speed but is free to pitch about its center of gravity, the equation of motion may be derived from Newton's second law, as described above. For pure pitching motion, the sum of the moments about the center of gravity is equal to the product of the moment of inertia about the local y-axis ( $I_y$ ) and the angular acceleration about the y-axis ( $\ddot{\theta}$ ).

$$\Sigma M_{cg} = I_y \ddot{\theta} \quad (2.20)$$

If the moment and angle are treated as the sum of a reference value and a perturbation as, for example,  $M = M_0 + \Delta M$ , and the reference condition is the trimmed state, when the moment is zero, then Equation 2.20 becomes

$$\Delta M_{cg} = I_y \Delta \ddot{\theta} \quad (2.21)$$



Considering only pitching motion, the equation of motion is a function of the angle of attack ( $\alpha$ ), the time rate of change of the angle of attack, denoted by a dot, ( $\dot{\alpha}$ ), the time rate of change of the pitch angle ( $\dot{\phi} = q$ ), and the elevator angle ( $\delta_e$ ). Using a first order Taylor series expansion, the change in pitching moment may be expressed as

$$\Delta M = \frac{\partial M}{\partial \alpha} \Delta \alpha + \frac{\partial M}{\partial \dot{\alpha}} \Delta \dot{\alpha} + \frac{\partial M}{\partial q} \Delta q + \frac{\partial M}{\partial \delta_e} \Delta \delta_e \quad (2.22)$$

where  $q$  is the time rate of change of the pitching angle.

Because the center of gravity is constrained, if the fixed frame of reference is initially aligned with the body-fixed frame then  $\Delta \alpha = \Delta \theta$ ,  $\Delta \dot{\alpha} = \Delta \dot{\theta} = \Delta q$ , and  $\Delta \ddot{\theta} = \Delta \dot{q}$ . Substituting these relationships into Equations 2.21 and 2.22 and re-writing the aerodynamic derivatives gives

$$\Delta \ddot{\alpha} - (M_q + M_{\dot{\alpha}}) \Delta \dot{\alpha} - M_{\alpha} \Delta \alpha = M_{\delta_e} \Delta \delta_e \quad (2.23)$$

where

$$M_{\alpha} = \frac{\partial M}{\partial \alpha} / I_y \quad M_{\dot{\alpha}} = \frac{\partial M}{\partial \dot{\alpha}} / I_y \quad \text{and so on} \quad (2.24)$$

In Equation 2.23,  $M_q + M_{\dot{\alpha}}$  is the pitch damping derivative sum and  $M_{\alpha}$  is the static stability derivative. Equation 2.23 is a nonhomogeneous second-order differential equation with constant coefficients. Setting  $\Delta \alpha = Ae^{\lambda t}$ , the homogeneous portion of this equation of motion becomes

$$\lambda^2 Ae^{\lambda t} - (M_q + M_{\dot{\alpha}}) \lambda Ae^{\lambda t} - M_{\alpha} Ae^{\lambda t} = 0 \quad (2.25)$$

Dividing by  $Ae^{\lambda t}$  yields the characteristic equation of motion for Equation 2.25:

$$\lambda^2 - (M_q + M_{\dot{\alpha}}) \lambda - M_{\alpha} = 0 \quad (2.26)$$

The roots of the characteristic equation are known as the eigenvalues of the system. Applying the quadratic formula, the eigenvalues of this system may be shown to be [18]

$$\lambda_{1,2} = \frac{(M_q + M_{\dot{\alpha}})}{2} \pm \sqrt{\left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2 + M_{\alpha}} \quad (2.27)$$

The general form of the homogeneous part of a solution to a second-order equation is:  $\alpha(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}$ , where  $C_1$  and  $C_2$  are constants based on the initial conditions of the problem [6]. The actual response to a displacement from equilibrium depends on the eigenvalues, and the eigenvalues are based on the physical stability derivatives  $M_{\alpha}$ ,  $M_{\dot{\alpha}}$ , and  $M_q$ . In particular, the value of Equation 2.27's determinant dictates the response of the system.

If  $-M_{\alpha} < \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2$ , the eigenvalues of the system are real, and as long as they are both negative, the motion dies out exponentially with time. This is a condition known as overdamping. The equation of motion for this case becomes

$$\begin{aligned} \alpha(t) = & C_1 \exp \left[ \left( \frac{M_q + M_{\dot{\alpha}}}{2} + \sqrt{\left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2 + M_{\alpha}} \right) t \right] \\ & + C_2 \exp \left[ \left( \frac{M_q + M_{\dot{\alpha}}}{2} - \sqrt{\left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2 + M_{\alpha}} \right) t \right] \end{aligned} \quad (2.28)$$

When  $-M_{\alpha} > \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2$ , the roots are complex, and the equation of motion is

$$\alpha(t) = \exp\left(\frac{M_q + M_{\dot{\alpha}}}{2}t\right) \left[ C_1 \exp\left(i\sqrt{-M_\alpha - \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2}t\right) + C_2 \exp\left(-i\sqrt{-M_\alpha - \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2}t\right) \right] \quad (2.29)$$

which may be simplified to [6]

$$\alpha(t) = \exp\left(\frac{M_q + M_{\dot{\alpha}}}{2}t\right) \left[ A \cos\left(\sqrt{-M_\alpha - \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2}t\right) + B \sin\left(\sqrt{-M_\alpha - \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2}t\right) \right] \quad (2.30)$$

The response defined by Equation 2.30 behaves as a damped sinusoid with a natural frequency,  $\omega$ , of

$$\omega = \sqrt{-M_\alpha - \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2} \quad (2.31)$$

The final case represents the boundary between exponential damping and sinusoidal damping and is known as critical damping. Critical damping occurs when  $-M_\alpha = \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)^2$  and the eigenvalues are repeated as  $\lambda_{1,2} = \left(\frac{M_q + M_{\dot{\alpha}}}{2}\right)$ . For the case of repeated roots, the form of the equation of motion is [6]

$$\alpha(t) = (C_1 + C_2 t) e^{\lambda t} \quad (2.32)$$

and, if  $\lambda < 0$ , the exponential term goes to zero faster than  $C_2 t$  goes to infinity with time, and the motion damps out without oscillations.

The damping for the critically damped case is known as the critical damping, and is defined as the value that makes the determinant of Equation 2.27 equal to zero:

$$-(M_q + M_{\dot{\alpha}})_{cr} = 2\sqrt{-M_{\alpha}} \quad (2.33)$$

The damping of any oscillatory case is typically defined in terms of a damping ratio,  $\zeta$ , which is

$$\zeta = \frac{-(M_q + M_{\dot{\alpha}})}{2\sqrt{-M_{\alpha}}} \quad (2.34)$$

When there is no damping in the system,  $\zeta = 0$  and the system oscillates in a constant sine wave. The natural frequency for this special case is called the undamped natural frequency  $\omega_n$

$$\omega_n = \sqrt{-M_{\alpha}} \quad (2.35)$$

It is helpful to refer these results to the standard form of a second-order differential equation with constant coefficients, which defines the motion for the generic second order system in Equation 1.2. This equation is:

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0 \quad (2.36)$$

the roots of which are

$$\lambda_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1 - \zeta^2} \quad (2.37)$$

The damping of the system is governed by the real part of these complex roots, and the damped natural frequency is governed by the imaginary part. As an example, Figure 2.6 shows the response of a generic second order system to a step input when the natural frequency is held constant and the damping ratio is varied.

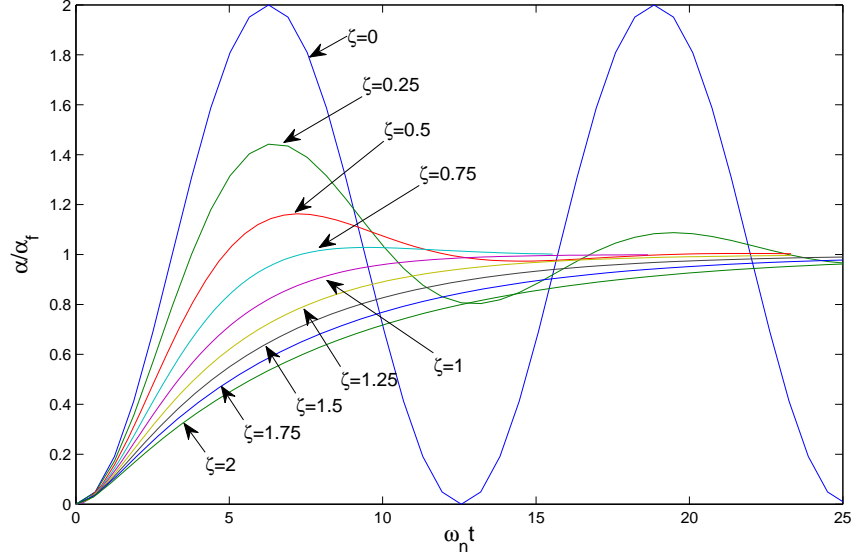


Figure 2.6: Effect of pitch damping on pure pitching response.

The response is defined by Equation 2.38,

$$\frac{\alpha}{\alpha_f} = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin\left(\omega_n t \sqrt{1-\zeta^2} + \phi\right) \quad (2.38)$$

where  $\alpha_f$  is the final value of  $\alpha$  and  $\phi$  is the phase angle defined as:

$$\phi = \sin^{-1}\left(\sqrt{1-\zeta^2}\right)$$

A full derivation of Equation 2.38 may be found in references [18, 20, 35]. Note the undamped sinusoid for the  $\zeta = 0$  case and the decrease in overshoot as  $\zeta$  increases.

*2.1.3 Pure Rolling Motion.* In a manner analogous to the methods for pure pitching, the equations of motion for pure rolling may be derived from Newton's second law

$$\Sigma \text{RollingMoments} = I_x \ddot{\phi} \quad (2.39)$$

The rolling moments may be due either to aileron deflection or rolling rate, so Equation 2.39 may be re-written as

$$\frac{\partial L}{\partial \delta_a} \Delta \delta_a + \frac{\partial L}{\partial p} \Delta p = I_x \Delta \ddot{\phi} \quad (2.40)$$

Since the rolling rate is equal to the time rate of change of the rolling angle,  $p = \dot{\phi}$ , Equation 2.40 becomes

$$\Delta \ddot{\phi} - L_p \Delta \dot{\phi} = L_{\delta_a} \Delta \delta_a \quad (2.41)$$

where  $L_p$  and  $L_{\delta_a}$  are defined as

$$L_p = \frac{\partial L / \partial p}{I_x} \quad \text{and} \quad L_{\delta_a} = \frac{\partial L / \partial \delta_a}{I_x} \quad (2.42)$$

This is again a second order nonhomogeneous equation with constant coefficients, but, unlike the pitching equation, there is no stiffness term, only damping. The characteristic equation of this nonhomogeneous equation is

$$\lambda^2 - L_p \lambda = 0 \quad (2.43)$$

and the eigenvalues are

$$\lambda_{1,2} = 0, L_p \quad (2.44)$$

where  $L_p$  is the roll damping derivative. These eigenvalues are both real, so the response is overdamped for  $L_p < 0$ , and the form of the resulting equation of motion is

$$\phi(t) = C_1 e^{L_p t} + C_2 \quad (2.45)$$

$L_p$  must be negative for a dynamically stable system. If the roll damping is negative, the first term of Equation 2.45 goes to zero as time increases and the roll angle goes to the constant  $C_2$ , which is based on initial conditions and inputs. The magnitude of the  $L_p$  term will determine the speed with which the constant roll angle is reached. Large negative values for roll damping will lead to rapid responses; small negative values will cause the system to respond more slowly. Positive values for the roll damping derivative will cause the system to roll uncontrollably and diverge.

## ***2.2 Governing Equation***

The Beggar code is a finite volume, cell-centered flow solver with the ability to solve for a large variety of cases: inviscid or viscous, steady or turbulent, and static or moving. One of Beggar’s key capabilities is that it can resolve the flow around multiple complex bodies in relative motion. This capability is primarily enabled by two methods: the ability to allow for blocked, patched, and overset grid communication and a 6+ degree-of-freedom solver that uses the forces and moments computed by the flow solver to predict the motion of rigid bodies [2]. Beggar combines these methods and, when the correct cases are run, may be used to produce the time-accurate output necessary for the computation of static and dynamic stability derivatives.

In the present analysis, the Beggar code was used to solve the Euler equations for inviscid cases and the Reynolds-Averaged Navier-Stokes equations for viscous cases. The Navier-Stokes equations are presented below, and the Euler equations may be obtained by removing the viscous terms. The Reynolds-Averaged Navier-Stokes Equations are found as the time-average of the Navier-Stokes equations [4].

The Navier-Stokes (N-S) equations are based on the conservation of mass, momentum, and energy. The Beggar code solves these equations using a finite volume solver, which requires that the N-S equations be in integral form, shown below as a single vector equation [15]:

$$\int_V \frac{\partial \vec{Q}}{\partial t} dV + \int_A (\vec{F}_c - \vec{F}_v) dA = 0. \quad (2.46)$$

In Equation 2.46, body and source terms have been neglected, and  $\vec{Q}$ ,  $\vec{F}_c$ , and  $\vec{F}_v$  are the vectors of conserved variables, convective fluxes, and viscous fluxes respectively. The vector of conserved variables is defined as:

$$\vec{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{bmatrix} \quad (2.47)$$

$E_t$  is the total energy, defined as:

$$E_t = \rho \left( e + \frac{1}{2} |\vec{V}|^2 \right) \quad (2.48)$$

where  $e$  is the internal energy.

The convective flux term accounts for the inviscid fluxes and flow terms acting on a control volume, and is defined as:

$$\vec{F}_c = \begin{bmatrix} \rho \tilde{U} \\ \rho u \tilde{U} + p n_x \\ \rho v \tilde{U} + p n_y \\ \rho w \tilde{U} + p n_z \\ (E_t + p) \tilde{U} \end{bmatrix} \quad (2.49)$$

where  $n_x$ ,  $n_y$ , and  $n_z$  are unit vectors normal to the control volume and the term,  $\tilde{U}$ , is known as the contravariant velocity, which is:



$$\tilde{U} = \vec{V} \cdot \hat{n} = un_x + vn_y + wn_z \quad (2.50)$$

The viscous term,  $\vec{F}_v$  accounts for both the viscous fluxes acting in all directions and the conduction terms due to work and heat:

$$\vec{F}_v = \begin{bmatrix} 0 \\ \tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z \\ \tau_{yx}n_x + \tau_{yy}n_y + \tau_{yz}n_z \\ \tau_{zx}n_x + \tau_{zy}n_y + \tau_{zz}n_z \\ \Theta_x n_x + \Theta_y n_y + \Theta_z n_z \end{bmatrix} \quad (2.51)$$

The work terms and heat conduction terms of Equation 2.51 ( $\theta_i$ ), are of the form:

$$\theta_x = u_x \sigma_{xx} + v \tau_{xy} + w \tau_{xz} + k \frac{\partial T}{\partial x} \quad (2.52)$$

and

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \nabla \vec{V} \quad (2.53)$$

is the viscous stress tensor for a Newtonian fluid. Assuming Stoke's Hypothesis, ( $\lambda + \frac{2}{3}\mu = 0$ ), these terms become:

$$\tau_{ij} = \tau_{ji} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \nabla \vec{V} \right) \quad (2.54)$$

In order to put Equation 2.46 into Euler equation form, it is assumed that the viscous terms are negligible, and they are treated as zero. Strictly speaking, the Euler equations only include the momentum portion of Equation 2.46 with these assumptions, but the mass and energy equations are often included as well. With the source terms (body forces, body heating, mass injection, etc...) again treated as zero, the integral form of the Euler equation is [15]:

$$\int_V \frac{\partial \vec{Q}}{\partial t} dV + \int_A (\vec{F}_c) dA = 0. \quad (2.55)$$

Beggar uses dimensional values to calculate actual forces and moments acting on a body, but it uses non-dimensional values to solve the Navier-Stokes equations. Non-dimensionalization enables Beggar to find a solution for a given model and then apply that same solution to multiple cases with different dimensional parameters, but the same dimensionless parameters. Each of the flow variables is made dimensionless according to Equation 2.56. The dimensionless parameters are indicated by the asterisks and are used to replace the dimensional values in the N-S equations.

$$\begin{aligned} \rho^* &= \rho/\rho_\infty & E_t^* &= E_t/\rho_\infty a_\infty^2 & p^* &= p/\rho_\infty a_\infty^2 & t^* &= ta_\infty/L_{ref} \\ u^* &= u/a_\infty & v^* &= v/a_\infty & w^* &= w/a_\infty \end{aligned} \quad (2.56)$$

### 2.3 Solver Methods

As stated above, Beggar uses a finite volume, cell-centered approach to flow solving. While the flow is governed by the Navier-Stokes equations at all points and times, the equations cannot be solved for the flow directly because of the difficulty involved in solving non-linear, partial differential equations [36]. In order to apply these equations to practical, non-simplified problems, the equations must be discretized so that numerical approximations may be obtained.

Beggar is capable of both implicit and explicit time discretization, but an implicit solver is more effective because it maintains solution accuracy and stability with larger timesteps [4]. Equation 2.57 shows an implicit discretization of Equation 2.46:

$$\frac{\partial Q^{n+1}}{\partial t} V + \sum_{faces} \left( \vec{F}_c^{n+1} - \vec{F}_v^{n+1} \right) = 0 \quad (2.57)$$

where  $\frac{\partial Q^{n+1}}{\partial t}$  is the temporal discretization of the change in the conserved variables. This term multiplied by the cell volume must be equal to the sum of the fluxes through the boundaries of the cell. The fluxes are linearized in time as [15]

$$F_c^{n+1} \approx F_c^n + \frac{\partial F_c}{\partial Q} (Q^{n+1} - Q^n) \quad (2.58)$$

$$F_v^{n+1} \approx F_v^n + \frac{\partial F_v}{\partial Q} (Q^{n+1} - Q^n) \quad (2.59)$$

Substituting these linearized fluxes into Equation 2.57 yields:

$$\frac{\partial Q^{n+1}}{\partial t} V + \frac{\partial R}{\partial Q} (Q^{n+1} - Q^n) = -R^n \quad (2.60)$$

where  $\frac{\partial R}{\partial Q}$  is the flux Jacobian, defined as:

$$\frac{\partial R}{\partial Q} = \sum \left( \frac{\partial F_c^n}{\partial Q} + \frac{\partial F_v^n}{\partial Q} \right) \quad (2.61)$$

The right hand side of Equation 2.60 is the explicit side, known as the residual, which is defined as:

$$R^n = \sum (F_c^n - F_v^n) \quad (2.62)$$

Two options are available in Beggar for the implicit time discretization term: an Euler time discretization and a second order three point backward time discretization. The three point backward time discretization is the more accurate of the two, and is given by:

$$\frac{\partial Q^{n+1}}{\partial t} = \frac{3Q^{n+1} - 4Q^n + Q^{n-1}}{2\Delta t} \quad (2.63)$$

For unsteady solutions, it is essential that the solution be sufficiently converged at every timestep in order to maintain accuracy. Beggar uses Newton iterations to guarantee appropriate convergence for each timestep by allowing the user to specify either the number of Newton iterations or a convergence criteria. For a vector function  $G(x) = 0$ , Newton's method can be written as [37]:

$$G'(x^m)(x^{m+1} - x^m) = -G(x^m) \quad (2.64)$$

where  $G'(x)$  is the Jacobian matrix:

$$G'(x) = \begin{bmatrix} a_{11}(x) & a_{12}(x) & \cdots & a_{1,n}(x) \\ a_{21}(x) & a_{22}(x) & \cdots & a_{2,n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(x) & a_{n2}(x) & \cdots & a_{n,n}(x) \end{bmatrix} \quad (2.65)$$

where

$$a_{ij}(x) = \frac{\partial G_i(x)}{\partial x_j} \quad (2.66)$$

Assuming Euler discretization, equation 2.60 may be rearranged to resemble 2.64:

$$\left(\frac{\partial R}{\partial Q}\right)^{n,m} (Q^{n+1,m+1} - Q^{n+1,m}) = - \left[ V \frac{Q^{n+1} - Q^n}{\Delta t} + R^n \right] \quad (2.67)$$

Solving this equation directly would require the inversion of the Jacobian term, which may have dimensions on the order of millions. Instead, Beggar applies a symmetric Gauss-Seidel method, which solves the generic equation  $[A]x = b$  for  $x$  by dividing  $[A]$  into

$$[A] = ([D][L])[U] \quad (2.68)$$

where  $D$  is the diagonal part of  $A$ , and  $U$  and  $L$  are the parts of  $A$  above and below the diagonal, respectively.  $x$  is found by solving

$$x_i^k = \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^k - \sum_{j=i+1}^{j_{max}} A_{ij} x_j^{k-1} \right) / A_{ii} \quad (2.69)$$

The Gauss-Seidel iterations are referred to as inner iterations within Beggar and are run until user-specified values for convergence or maximum inner iterations are reached.

*2.3.1 Boundary Conditions.* Properly defined boundary conditions are critical to accurately simulating the flow about an object. The default boundary condition within Beggar is an inviscid boundary, termed “tangent.” At a tangent boundary, the component of the velocity normal to a surface is set to zero, but the velocity magnitude does not decrease as the surface is approached, the flow is simply turned. For viscous solutions, the “no-slip” boundary condition may be used to decrease the flow velocity to zero as the surface is approached. Farfield boundaries are solved using characteristic boundary conditions. For supersonic cases, these boundary conditions are easily specified because all waves run downstream [2].

*2.3.2 Overset Grids.* As stated previously, Beggar is a structured solver. Three dimensional structured grids require that each cell have eight points and six faces. Structured grids easily map from computational to physical domains, but the process of grid generation for complex geometries can be challenging. To simplify the requirements for the grid generation process, multiple types of grid communication are available, as shown in Figure 2.7.

The highest level of Beggar’s grid hierarchy is the superblock, which consists of one or more related grids that do not overlap each other. Within a superblock, the only permissible types of communication between separate grids are block-to-block and patched, each illustrated in Figure 2.7. An overset, or chimera, grid assembly

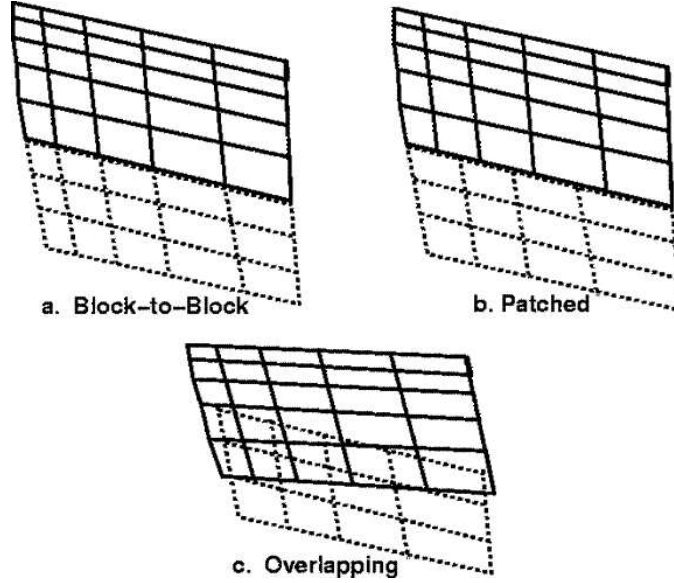


Figure 2.7: Beggar communication types between grids. [2]

allows superblocks to interpolate solutions from one another through overlapping communication. Values are interpolated to interior cell centers of each superblock, and the interpolation process requires an overlap of five cells for a first order stencil, seven cells for a second order stencil [15].

The overset assembly process is also used to cut holes in one superblock when another superblock has a solid surface at the same location. This capability allows grids to be constructed about a localized geometry without necessarily being concerned about other surfaces in the flow. Care must still be taken to allow for appropriate overlap and to ensure that cells interpolate from other cells of a similar size, but the Chimera process greatly simplifies three dimensional structured grid construction.

*2.3.3 Six Degree of Freedom Model.* Beggar's (6+)DOF capability allows the user to specify various types of motion. The true power of the solver lies in the coupling of the rigid body equations of motion and the CFD governing equations. This capability is applied by first solving the governing equations to find the forces and moments on a model and then using those forces and moments in the rigid body

equations of motion to determine an incremental response to the flow. This process is carried out in a time-accurate fashion to simulate actual trajectories.

Other options that use just some aspects of the (6+)DOF solver include constrained motion and motion prescribed by an input file. For constrained motion, the forces and moments are calculated as normal, but the rigid body response to those forces and moments is limited. As an example, in constrained planar motion, the rigid body equations of motion are solved and a response is calculated, but a dot product of that response and a vector normal to the planar constraint is taken to cause any response outside of that plane to be zero. Prescribed motion allows the user to specify exactly how the model will move, and the model will not respond at all to the forces calculated by the CFD solver.

*2.3.3.1 Coordinate Systems.* At this point, a note must be made about the differences in convention between CFD and standard controls coordinate systems. CFD coordinate systems customarily define the flow direction to be the direction of the positive x-axis, as shown in Figure 2.8(a). The y-axis is then assumed to go up, and the z-axis is perpendicular to the x and y-axes and is positive in the direction of the left wing. When dealing with aircraft controls, on the other hand, the axes are typically defined as shown in Figure 2.8(b). The x-axis is positive in the direction of flight, the z-axis is down, and the y-axis points out of the right wing of the aircraft.

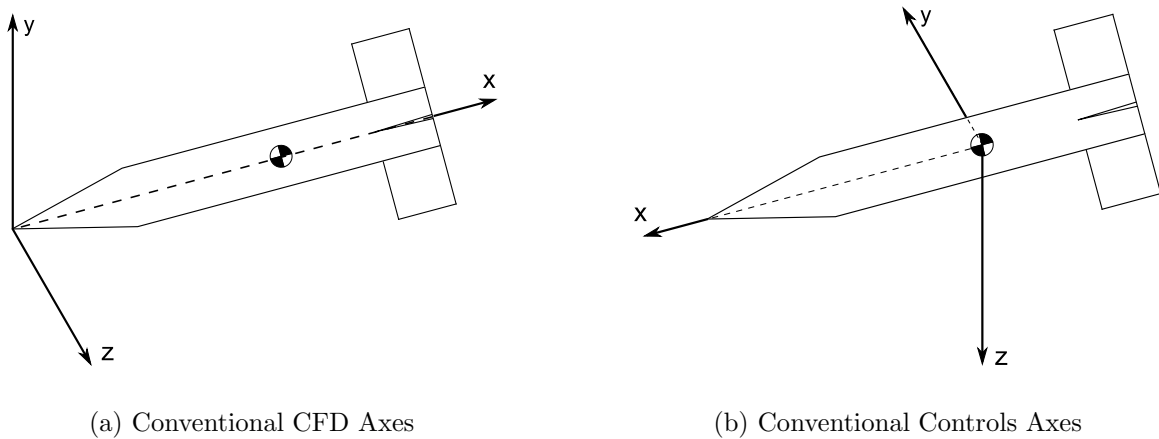


Figure 2.8: Comparison of conventional CFD and controls axes.

For consistency, the conventional controls axes will be used in the present analysis, except when referring to specific Beggar inputs.



### III. Methodology

Tests were performed on a commonly used missile configuration known as the Basic Finner. The static and dynamic pitching and rolling stability derivatives were determined by following the steps that are summarized below. These steps are described in more detail in this chapter.

1. Model and grid construction
2. Computation of static solutions
3. Computation of unsteady solutions using static solutions as initial conditions
4. Moment coefficient histories obtained from unsteady solutions
5. Calculation of stability derivatives from moment coefficient histories

#### ***3.1 Basic Finner Model and Geometry***

The Basic Finner model was chosen because of the extensive experimental and computational testing that has been performed with it in the past. Experimental data available include both wind tunnel data [33] and ballistic test range data [24,31]. Data available from computational fluid dynamics includes many different solvers and approaches [10,16,17,21,23,28].

*3.1.1 Model Geometry.* The Basic Finner is a slender body axisymmetric missile with four fins attached at the base. Figure 3.1 shows the dimensions of the missile in calibers, where the diameter is equal to one caliber. Each fin has a chord and height of one caliber, and a thickness of 0.08 calibers at the trailing edge. The leading edge of the fins comes nearly to a sharp edge, with a radius of 0.004 calibers. Similarly, the conical nose of the Basic Finner also has a radius of 0.004 calibers and has a half-angle of 10 degrees. The model center of gravity is located on the centerline, 6.1 calibers back from the nose.

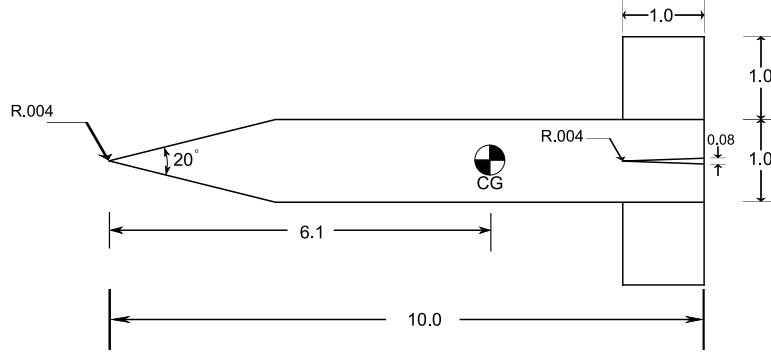


Figure 3.1: Basic Finner configuration (dimensions in calibers)

For the present analysis, the model was created using the commercially available software, Gridgen<sup>®</sup>. Figure 3.2 shows the completed Gridgen<sup>®</sup> model of the Basic Finner. The same model was used for both the inviscid and the viscous cases.

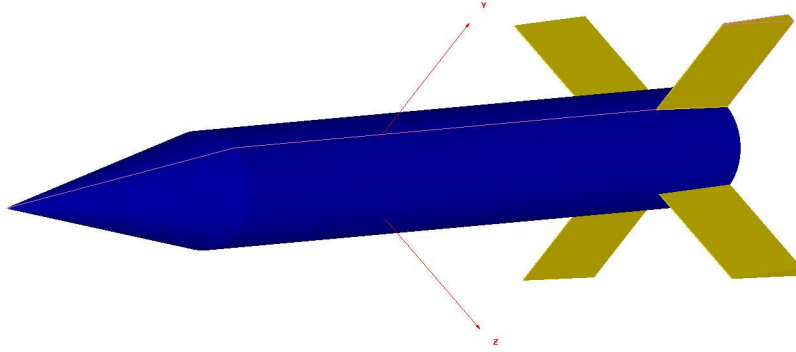


Figure 3.2: Three-dimensional Basic Finner model.

*3.1.2 Grid Generation.* The model of the Basic Finner was used with Gridgen<sup>®</sup> to create the structured grids required by Beggar. Because of Beggar's overset capabilities, described in Chapter II, it was possible to build separate grids

for the missile body and the fins. Building separate, relatively simple grids about each part of the geometry is desirable because attempting to build a single structured grid around the entire complicated geometry would require a large amount of time and effort to construct and refine.

Because the missile is divided into four identical quarters, it was possible to build all of the necessary grids on a single quarter and then rotate those grids around for the other sections. All of the grids around the body were combined into a single superblock structure, and the grids around each fin were grouped into individual superblocks, resulting in a total of five superblocks solving the flow around the missile. A sixth superblock was created as an inertial reference that does not move with the rest of the model. The purpose of the inertial grid is to provide a fixed reference point from which to determine the motion of the moving parts of the grid.

*3.1.2.1 Inviscid Grid.* The largest superblock created for the inviscid case was the body grid, which is the grid that goes around the body of the missile, ignoring the fins. The body grid was both the largest and the most coarse of the grids. It can accurately be considered the main grid, and all of the other grids operate with data passed to them from this main grid.

The body grid was divided into 3 blocks: one to solve in front of and above the missile, one to solve the flow immediately behind the missile (the 'plug' grid), and one to solve above the plug and behind the body. Since the flow over the missile was supersonic, the grid did not need to extend far in front of the body. To capture just some of what happens in front, the grid was extended half a missile length in front of the missile (one missile length is 10 calibers). Much more was expected to happen behind the missile, so the grid was extended a full missile length from the back of the missile so that the boundary would not affect the flow solution. To accurately capture the flow in the radial direction, the grid was extended one and a half missile lengths up. Figure 3.3 shows the three domains used to create these blocks.

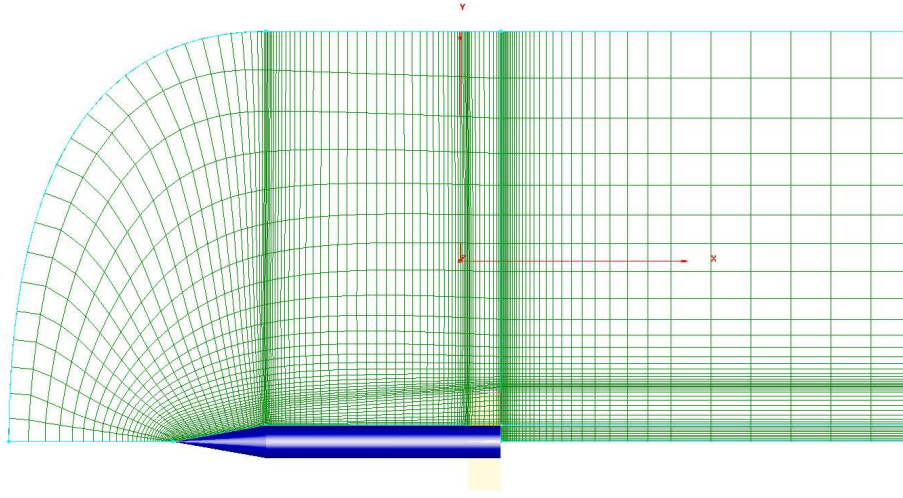


Figure 3.3: Inviscid body grid domain.

As Figure 3.3 shows, the largest degree of refinement in this grid was just above the nose and at the back near the fins. The purpose of the refinement at the nose was to capture any shocks that might occur. Oblique conical shocks were expected because of the supersonic speeds and the sharp point of the nose. The grid was refined near the fins for two reasons. First of all, the flow was expected to be affected by the fins, and this effect needed to be resolved. Secondly, the grids around the fins were more resolved, and, in order to enable accurate overset communication between the body and fin grids, the body grid needed to have spacing comparable to the spacing of the fin grid near the fins. For the inviscid case, the full  $360^\circ$  of the body superblock contained 505,920 cells.

The fin grid was created by first constructing domains on the surface of the fin, and then extruding those domains out in all directions, conforming to the shape of the missile body where appropriate. The blocks constructed included a C-grid around the sides and front of the fin, extending from the missile surface to above the top of

the fin, a block extending from the top of the fin to the top of the grid domain, and a block to capture the flow behind the fin. It was necessary to extend this rear block below the surface of the missile in order to maintain sufficient interpolation points in the wake flow. Figure 3.4 displays a rear view of one fin superblock.

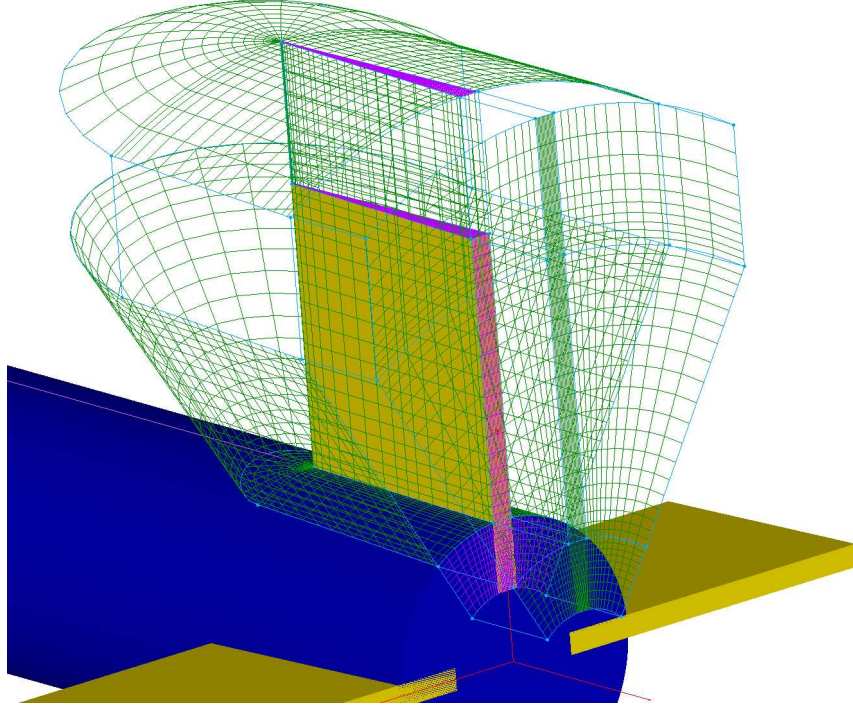


Figure 3.4: Inviscid fin grid superblock

The fin superblock was more refined than the body superblock in order to capture the effects of the flow around the fin. The refinement was relaxed further from the surface of the fin in order to achieve cell sizes appropriate for overset block communication with the body grid. Altogether, each inviscid fin superblock had 33,904 cells.

The inertial reference grid was created outside of the body and fin grids, and is simply a  $5 \times 5 \times 5$  block that does not move when the missile grids move. Figure 3.5 shows the inertial grid above the Basic Finner model with one quarter of the body grid and one fin grid assembled. With the body superblock, the four fin superblocks,

and the inertial block combined, table 3.1 shows the total number of cells for the inviscid case to be 641,661.

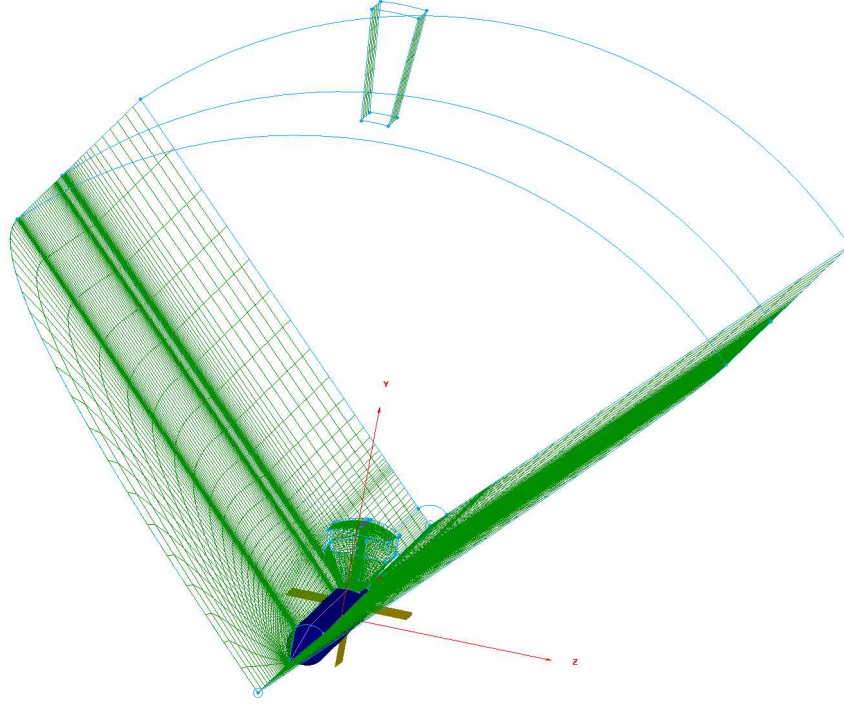


Figure 3.5: Basic finner block assembly

Table 3.1: Inviscid grid dimensions

Superblock	Number of Cells
Body	505,920
Fin	$4 \times 33,904$
Inertial	125
<b>Total</b>	<b>641,661</b>

*3.1.2.2 Viscous Grid.* The viscous grids were constructed in the same general manner as the inviscid grids, but additional care was necessary to ensure that initial grid spacing and growth rate satisfied the criteria for viscous grids. The boundary spacing was selected in order to achieve a  $y+$  value of 1.0, which corresponds to  $2.130 \times 10^{-4}$  calibers for this case [1] (Reynolds number based on diameter set to  $0.086 \times 10^6$  to match reference data [33]). From the missile body, the grids were

extended with a growth rate of 1.2 for 15 cells in the viscous boundary layer [15], and then the growth rate was increased to 1.3 for the remainder of the grid. Figure 3.6 shows all 360° of this body grid. Similar to the inviscid grid, the viscous body grid was also built with cells clustered near the nose and near the fins.

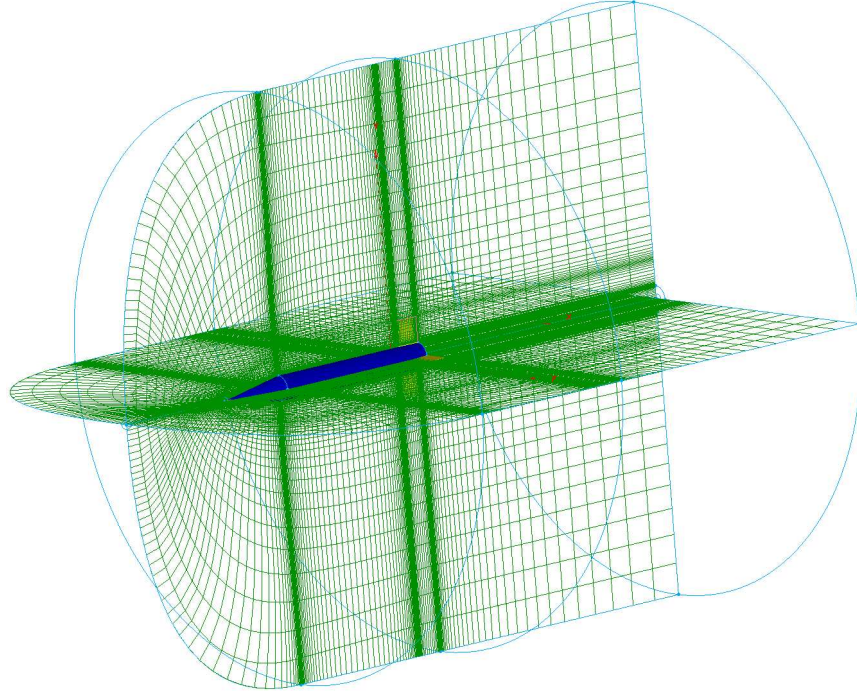


Figure 3.6: Viscous body superblock

The fin grids were constructed with the same initial spacing and growth rate as the body grid, and in the same shape as the inviscid fin grid. The same inertial grid was used for the viscous case as for the inviscid case. Table 3.2 shows that the total number of cells for the body grid was approximately 2.17 million and the total for each fin grid was about 0.3 million. This resulted in a total of approximately 3.4 million cells for the viscous case.

### 3.2 *Static Solutions*

In order to determine the static and dynamic stability coefficients for the Basic Finner model, dynamic testing was required. Each dynamic test was initialized from

Table 3.2: Viscous grid dimensions

Superblock	Number of Cells
Body	2,172,544
Fin	$4 \times 310,274$
Inertial	125
<b>Total</b>	3,413,765

a fully converged static solution, so static testing was also required. There were ten static cases of interest. Five were run with Mach number held constant at 1.96 while the angle of attack was varied from  $0^\circ$  to  $20^\circ$ , and five were run with angle of attack held constant at  $0^\circ$  while the Mach number was varied from 1.58 to 2.50. These ranges of Mach number and angle of attack were chosen based upon the availability of comparison data. Table 3.3 shows a complete listing of the static cases.

Table 3.3: Flow and model parameters for static cases.

Constant	Varied
$M = 1.96$	$\alpha = [0 \ 5 \ 10 \ 15 \ 20]^\circ$
$\alpha = 0^\circ$	$M = [1.58 \ 1.75 \ 1.89 \ 2.10 \ 250]$

All ten of the static cases shown in Table 3.3 were run for the inviscid grid using Beggar. Only the five cases at constant Mach number and varying angle of attack were run for the viscous grid.

In order to achieve convergence, each case was run until residuals had been reduced by at least three orders of magnitude, which is generally considered sufficient for engineering applications. Figure 3.7 shows a characteristic residual plot. This example is for the inviscid  $M = 1.96$ ,  $\alpha = 5^\circ$  case. Most of the cases converged quite readily from freestream startup values with moderate timestep ramping. Timestep ramping is the process of gradually increasing the size of the timestep from a small value to a value that will converge more rapidly. Initial timesteps must be small for stability at the startup, which is an issue because of the large initial gradients. Initial nondimensional timesteps were on the order of 0.0001 and reached a value of 1.0 in



500 iterations or less. For these cases, the desired convergence was achieved within around 1,000 iterations.

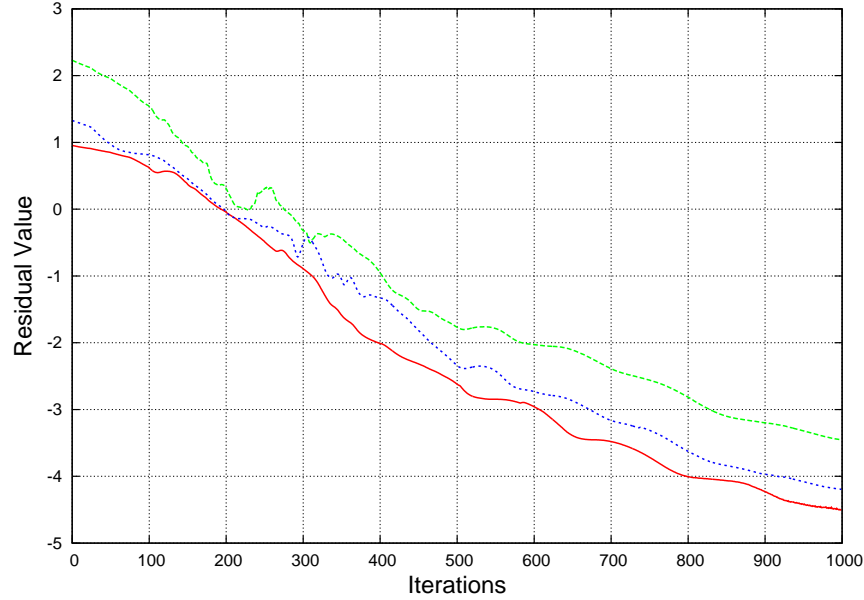


Figure 3.7: Sample residual plot:  $M = 1.96$ ,  $\alpha = 5^\circ$

For the two cases that were the most extreme,  $M = 1.96$ ,  $\alpha = 20^\circ$  and  $M = 2.50$ ,  $\alpha = 0^\circ$ , convergence was slightly more difficult to achieve. For each of these cases, the solver developed negative values for pressure or density in the base flow region because of the large gradients that existed immediately after the case began. In both cases, this issue was fixed by extending the timestep ramps and initializing from the  $M = 1.96$ ,  $\alpha = 0^\circ$  case. Extending the timestep ramp caused the solver to take smaller steps at the start and developed less extreme gradients. Initializing from previous solutions also gave the solver realistic values in the base flow region so that large gradients did not occur. Each of these solutions converged in around 1,100 iterations.

### 3.3 Dynamic Pitch

Dynamic pitch derivatives were obtained using two methods of oscillation about the model center of gravity. One method used prescribed sinusoidal forced oscillation

to determine the stability derivatives at one angle of attack per test. The other method allowed the model to oscillate freely in the pitch direction and was used to find the stability derivatives over a range of angles of attack. Each of the dynamic pitch cases was initialized from a static solution before the motion started.

For the cases tested using forced oscillation, the prescribed motion of the model was defined as:

$$\alpha = \alpha_o + \alpha_m \sin(\omega t) \quad (3.1)$$

where  $\alpha$  is the instantaneous angle of attack,  $\alpha_o$  is the starting and mean angle of attack,  $\alpha_m$  is the magnitude of oscillation, and  $\omega$  is the frequency of oscillation.

The parameters chosen to define this motion were the amplitude of oscillation and a nondimensional pitch rate,  $k_q$ , which is known as the reduced pitch rate and is defined as

$$k_q = \frac{qd}{2V_\infty} \quad (3.2)$$

where  $q$  is the pitch rate,  $d$  is the model diameter, and  $V_\infty$  is the freestream velocity of the flow. Use of a nondimensional pitch rate is beneficial because it enables comparison of cases of differing Mach number and model size. From the choices of  $\alpha_m$  and  $k_q$ , the frequency of oscillation is defined as

$$\omega = \frac{q}{\alpha_m} = \frac{2k_q V_\infty}{d\alpha_m} \quad (3.3)$$

In all cases, dimensional values were used for the model diameter and the freestream velocity. The model diameter was 1.25 inches, and the freestream velocity was the product of the Mach number and the speed of sound, which is based on altitude. The tests were run simulating an altitude of 20,000 ft, where the speed of sound is 1037 ft/s.

The reduced pitch rate and oscillation amplitude fully defined the motion of the forced oscillation cases, but other parameter choices were also vital to accurately resolving the flow. These parameters include the number of Newton iterations the solver used, the total number of oscillations, and the number of iterations per oscillation. The number of Newton iterations was important because time-accurate flow solving requires that the inner iterations be fully converged at each timestep. Multiple oscillations were required because the dynamic cases exhibit transient solutions upon startup. Additional oscillations removed those transients, and the solution eventually settled into a repeating cycle. Finally, the number of iterations per oscillation (which is inversely related to the timestep) can have a profound effect upon a solution. Too few iterations, and the timestep will be too large, and the flow under-resolved. Inaccurate force and moment data may then be obtained.

It is theoretically possible to simply set the flow parameters controlling convergence (Newton iterations, number of oscillations, and number of iterations per oscillation) to very high numbers to ensure that the solution is converged, but the associated computational expense would be alarming. For this reason, convergence studies were performed on each of these parameters to ensure both solution accuracy and reasonable use of computational resources. The convergence studies were run by starting each of these parameters at a minimal value, and successively increasing them until the measured stability derivatives no longer changed as the parameters increased. A change of less than 2% was required for convergence.

Convergence studies were also performed on the reduced pitch rate and amplitude of oscillation, although the studies for these were not so linear. The reduced pitch rate was varied to determine its effect upon the stability coefficients, and to assist in choosing a value that both ensured convergence and avoided nonlinear separation effects due to high angular velocities.

The amplitude of oscillation was important for a number of reasons. First of all, for a given reduced pitch rate, an oscillation of larger amplitude takes a longer amount

of time. This means that larger amplitudes require either a larger timestep, which may degrade accuracy, or more iterations, which requires additional computational time. Large amplitudes are also not advantageous because the local stability derivatives are desired, and large oscillations would smooth out the local effects, providing an average over a range rather than a local value.

Oscillations may not be too small, however, because hysteresis effects could dominate the solution, rather than the expected physics. Hysteresis effects occur when there is a lag between the cause and effect of a system. In a pitching missile system, this lag would be observed between the motion of the missile and the response of the flow; the flow responding only after the missile moved, rather than as it moved. Hysteresis effects are typically observed at low amplitudes of oscillation [22], so the amplitude at which tests were run needed to be high enough to avoid this.

The desired result was a range of values for which the solution was not significantly changed by local changes in the reduced frequency or amplitude. This was desired in order to determine solutions that were relatively independent of the parameters chosen. The parameters were varied both separately and together, and with a varying number of timesteps, and final values were chosen in the middle of what was determined to be the optimal range of the values tested.

The specific execution of the convergence studies is discussed in Chapter IV, along with the results of those studies. Using those results, ten prescribed motion dynamic pitch cases were tested using the inviscid grid, and five with the viscous grid. Both inviscid and viscous solutions were run for five cases with constant Mach number equal to 1.96. In these five cases, the angle of pitch about which the oscillation took place varied from  $0 - 20^\circ$  in  $5^\circ$  increments, as shown in the first part of Table 3.4. The other five cases, those oscillating about an angle of attack of  $0^\circ$  with the Mach numbers shown in the second part of Table 3.4, were run only on the inviscid grids.

In addition to the ten forced oscillation cases run on the inviscid grids, four inviscid free oscillation cases were tested. Three were run with Mach number equal to

Table 3.4: Dynamic test cases

Constant	Varied
$M = 1.96$	$\alpha = [0 \ 5 \ 10 \ 15 \ 20]^\circ$
$\alpha = 0^\circ$	$M = [1.58 \ 1.75 \ 1.89 \ 2.10 \ 250]$

1.96, and were started at angles of attack of 5, 20, and 30°. The final dynamic pitch case tested was free oscillation starting at  $M = 1.58$ , initialized from a static solution at  $\alpha = 20^\circ$ .

### 3.4 *Dynamic Roll*

Similar to the dynamic pitch cases, each dynamic roll case was initialized from a static solution and put through prescribed motion. Rather than the sinusoidal oscillation used in the in the pitch direction, however, the prescribed motion was a constant angular rate in roll. This difference in method is due to a difference in the calculation of damping derivatives in pitch and roll that is explained below.

As was the case in the dynamic pitch cases, there are certain parameters that must be chosen for the dynamic roll cases. The first of these parameters was the reduced roll rate,  $k_p$ , defined as

$$k_p = \frac{pd}{2V_\infty} \quad (3.4)$$

where  $p$  is the roll rate. Just like the reduced pitch rate, the reduced roll rate enables direct comparison of cases with differing model sizes or Mach numbers.

The other parameter tested and determined was the number of iterations per revolution. Together, these two parameters define the rate of roll, the physical step size, and the temporal step size. Because their effect is interrelated, the two values were varied iteratively, testing multiple steps per revolution for every reduced roll rate, and vice versa.

An initial estimation of the reduced roll rate was chosen based on comparison data from Oktay and Akay [21], where  $k_p = 0.00326$  was used. The number of iterations per revolution, which is inversely related to the timestep, was tested in the same manner as iterations per oscillation for the pitch cases. The timestep convergence study began with just 360 iterations per revolution, and that value was doubled until increasing the number of iterations changed the solution by less than 2%. Using these methods, optimal parameters for both accuracy and speed were chosen.

The parameters defined by the methods outlined above were used for testing six dynamic roll cases.. All of these cases were run on the inviscid grid, and each of them was initialized from a static solution at zero degrees angle of attack, Mach numbers  $M = \begin{bmatrix} 1.58 & 1.75 & 1.89 & 1.96 & 2.10 & 2.50 \end{bmatrix}$ . Each case was run until the roll moment had converged.

### 3.5 *Beggar Inputs*

Each static and dynamic case required several different input files. The master input file, denoted by the extension “.in”, was used to set solver options like Mach number, angle of attack, CFL number, and solver type. It was also used to read in the (6+)DOF inputs and the grid inputs. Appendix A.1 shows an example of a “.in” file for the inviscid  $M = 1.96, \alpha = 5^\circ$  case. The inviscid solver used was a second order Euler solver with implicit boundaries that used Steger-Warming [4] methods to solve for the fluxes. The second order time discretization, three point backward time, was used for additional accuracy. The viscous cases were solved using a second order Baldwin-Lomax turbulence model [4]. The boundaries and viscous terms were implicitly updated, and Roe methods [4] were used to solve for the fluxes.

Files with the extension “.beg” were called from the master input file to read in the grids. Within each of these “.beg” files, a single grid was read in, boundary conditions for that grid were set, cell protection was enabled, if necessary, and local grid rotations were applied where appropriate. Located in Appendix A.1 is an example of a file used to read in the fin grids and rotate them  $90^\circ$  from the original location.

The force and dynamic group specifications, denoted by the extensions “.fspec” and “.dyn” were used to define the motion of the missile body. The missile body and fins were included in a single force specification to determine the total forces and moments on the model. The dynamic specification was used to define the prescribed motion or to specify the motion constraints. Example force and dynamic specifications are located in Appendix A.1, along with a sample of a file used to specify the prescribed motion.

### ***3.6 Stability Derivative Calculation***

Beggar outputs the solution time and the model location, orientation, forces, moments, and coefficients to user specified output files. The orientations and moment coefficients were extracted and analyzed using the **Matlab**<sup>®</sup> script found in Appendix A.2. The coefficient and orientation histories were then used to calculate the static and dynamic stability derivatives.

*3.6.1 Static Stability Derivatives.* Static stability derivatives are routinely found using both experimental techniques and CFD. The tests outlined above allow for three ways to calculate the static pitch stability coefficient,  $C_{m_\alpha}$ . First, the static solutions were used to plot  $\alpha$  vs  $C_m$ . The local slope of the curve at any angle of attack is the static pitch stability coefficient,  $C_{m_\alpha}$  [21].

The second method used the dynamic solutions from forced oscillation at each angle of attack to determine the local static pitch stability coefficient. First, the angle of attack and moment coefficient histories were plotted as  $C_m$  vs  $\alpha$ . The sinusoidal oscillation causes the plot of these histories to make a loop, as shown in Figure 3.8. To determine the static stability derivative from this plot, a line was drawn from the left-most point (lowest angle of attack) to the right-most point (highest angle of attack) through the center of the loop, which corresponds to the static moment coefficient. The slope of this line is the static pitch stability coefficient.

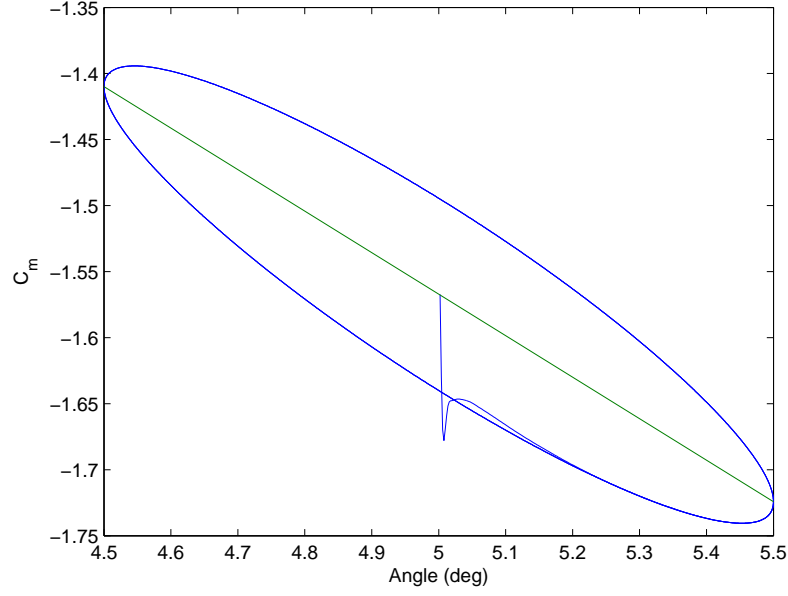


Figure 3.8: Sample moment coefficient history:  $M = 1.96$ ,  $\alpha = 5^\circ$

Finally, the static stability coefficient was also calculated from the free oscillation cases. This is best shown through an example. Figure 3.9(a) shows the trajectory history of a sample free oscillation case. In this example, the moment coefficient and the local pitch rate were extracted each time the model pitched through  $\alpha = 10^\circ$ . These moments were then plotted against the nondimensional pitch rate, as shown in Figure 3.9(b). A straight line was fit between the points on the curve, and the static moment coefficient ( $C_{m_{static}}$ ) at the angle of attack that the points were taken from was found as the y-intercept of that line. This method was employed to find  $C_{m_{static}}$  at numerous angles of attack along the trajectory, and then  $C_{m_\alpha}$  was found as the local slope of the  $C_{m_{static}}$  vs  $\alpha$  curve. The static pitch stability derivative relates to the static pitch stability coefficient as:

$$M_\alpha = \frac{QSd}{I_y} C_{m_\alpha} \quad (3.5)$$

Roll tests were done only for the cases with a pitch angle of zero. Because the Basic Finner is a symmetric missile, there is no static roll moment for zero angle of



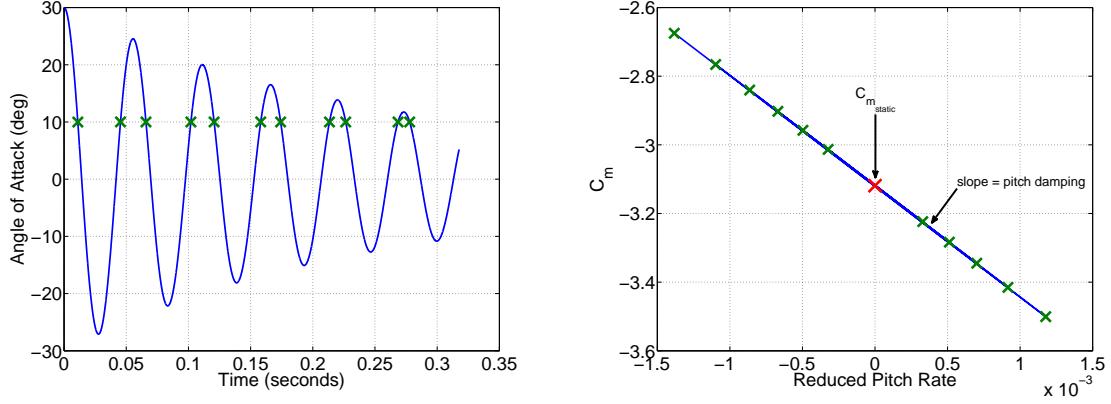


Figure 3.9: Free oscillation trajectory and method for determining pitch damping sum.

attack. This means that regardless of the roll angle, the static roll derivative,  $C_{l_\phi}$ , is zero when the angle of attack is zero. Thus, no static roll stability coefficients were calculated.

*3.6.2 Dynamic Stability Derivatives.* As shown in Chapter II, the dynamic pitch stability coefficient is  $(C_{m_q} + C_{m_{\dot{\alpha}}})$ . Because  $q = \dot{\alpha}$  for the missile in pure pitching motion, this sum may also be found from the forced oscillation pitching cycle shown in Figure 3.8 as

$$(C_{m_q} + C_{m_{\dot{\alpha}}}) = \frac{2V_\infty}{d} \frac{\partial C_m}{\partial q} = \frac{\partial C_m}{\partial k_q} \quad (3.6)$$

With forced oscillation, this equation was solved to determine the local pitch damping coefficient at each angle of attack by extracting the pitch moment coefficient at the angle of interest as the model oscillated up and down. As the model rotated up through the angle of interest, the reduced pitch rate had a value of  $k_q$ , and, as the model rotated back down, the reduced pitch rate was  $-k_q$ . For the example in Figure 3.8, the lower part of the curve corresponds to pitching up (positive  $k_q$ ) and the upper part corresponds to pitching down (negative  $k_q$ ). Using this information with Equation 3.6 yields

$$(C_{m_q} + C_{m_{\dot{\alpha}}}) = \frac{\Delta C_m}{2k_q} \quad (3.7)$$

where  $\Delta C_m = (C_{m_{pitchup}} - C_{m_{pitchdown}})$ .

The method for determining the pitch damping coefficient at each angle of attack from the free oscillation trajectories was very similar to the method for determining  $C_{m_\alpha}$ . As shown in Figure 3.9, the moment coefficients and rates were extracted at a given angle, and then the pitch damping coefficient was found as the slope of the straight line fit to the  $C_m$  vs  $\alpha$  plot [17]. The pitch damping derivative may be found from the pitch damping coefficient as

$$(M_q + M_{\dot{\alpha}}) = (C_{m_q} + C_{m_{\dot{\alpha}}}) \frac{QSd}{I_y} \frac{d}{2V_\infty} \quad (3.8)$$

The roll damping term was found as [21]

$$C_{l_p} = \frac{(C_l - C_{l_\phi} \phi)}{k_p} \quad (3.9)$$

Because the static roll stability term is equal to zero for a symmetric missile, this equation reduces to:

$$C_{l_p} = \frac{C_l}{k_p} \quad (3.10)$$

This roll damping coefficient may be re-written as the roll damping stability derivative:

$$L_p = C_{l_p} \frac{QSd}{I_y} \frac{d}{2V_\infty} \quad (3.11)$$

## IV. Results and Discussion

Using the geometry, grids, and methods developed in Chapter III, the static and dynamic stability coefficients were determined for the Basic Finner. Static cases at various angles of attack and Mach number were run first, both as a starting point for dynamic cases and as a comparison for the static stability coefficient. Dynamic cases were run with forced oscillation, free oscillation, and prescribed roll motion for the inviscid cases, and with forced oscillation for the viscous cases. The output from the dynamic cases was used to determine the static and dynamic stability derivatives and to characterize the disturbance response of the Basic Finner. A comparison of the results from the methods used here and from prior experimentation is included.

### 4.1 Static Solutions

*4.1.1 Inviscid.* Inviscid static tests were run for the ten cases shown in Table 3.3. As stated previously, each of these steady state solutions was run until the residuals had converged by at least three orders of magnitude. Figure 4.1 shows the residual convergence for the slowest case and for the case with the highest angle of attack. Each of the lines represents the average residual value for a given block in the grid, and the y-axis represents orders of magnitude of decrease in the residual value. The periodic spikes in the residual values are due to ramping of the flow solver

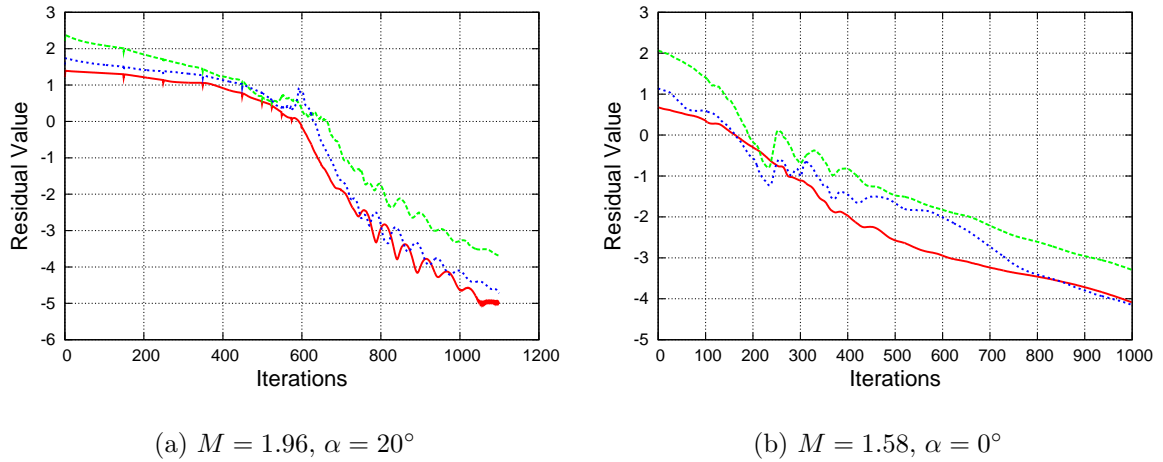
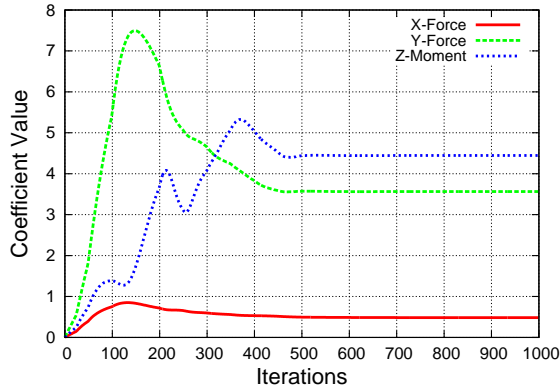


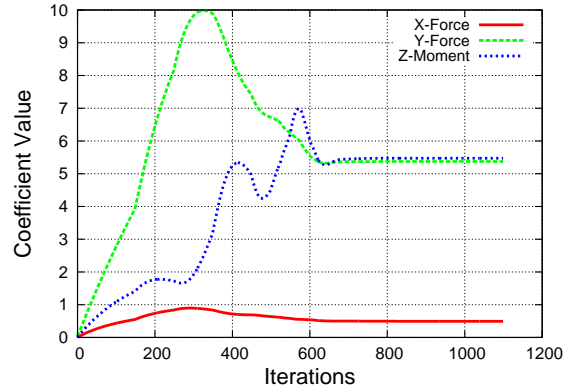
Figure 4.1: Inviscid residual convergence.

timestep. The residual convergence of each of the cases tested behaved in this same general manner.

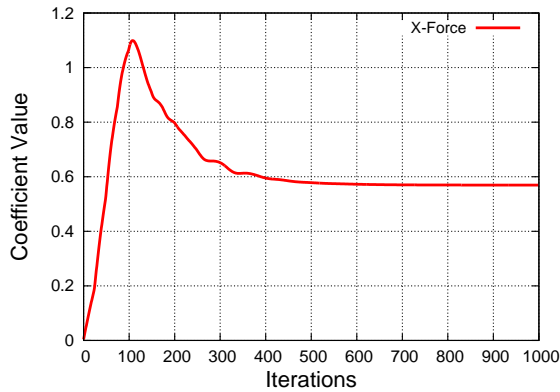
Equally important to the accuracy of steady state solutions is the convergence of the integrated forces and moments. Even when the residuals do not converge to the desired tolerance, it is often considered sufficient to monitor the force and moment histories to determine convergence. Figure 4.2 displays sample force and moment coefficient histories in CFD coordinates for cases with high angle of attack, low Mach number, and high Mach number. The force and moment coefficients for every case converged within the given number of iterations. As was the case for the residuals, the periodic spikes in the coefficients are due to the timestep ramping.



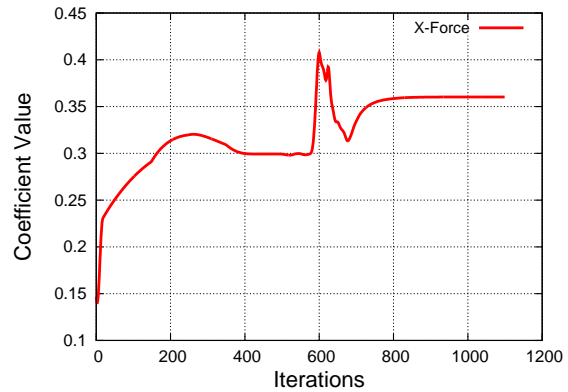
(a)  $M = 1.96, \alpha = 15^\circ$



(b)  $M = 1.96, \alpha = 20^\circ$



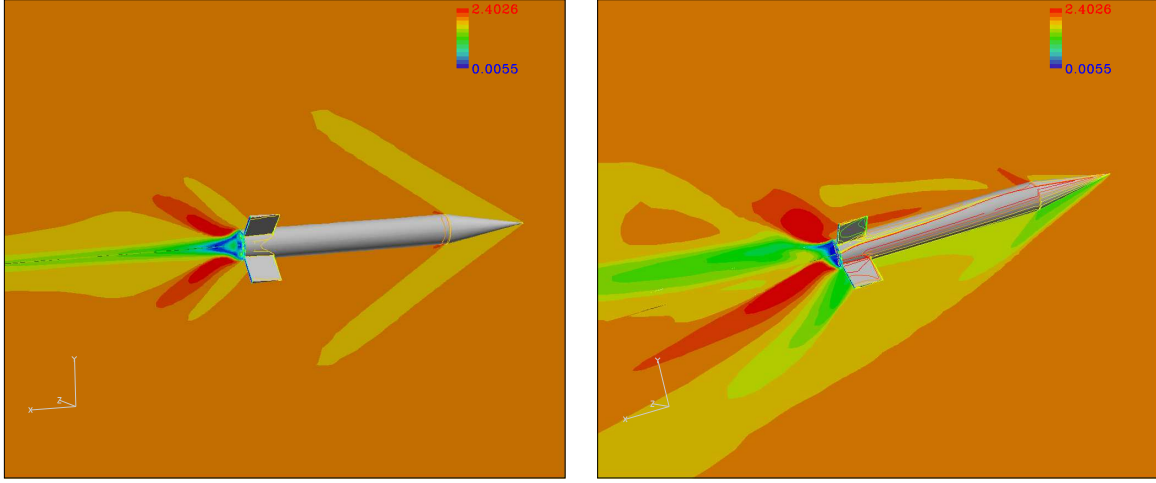
(c)  $M = 1.58, \alpha = 0^\circ$



(d)  $M = 2.50, \alpha = 0^\circ$

Figure 4.2: Inviscid force and moment coefficient histories.

To demonstrate that the flowfield of the static solutions is behaving as expected, sample results of static flow solutions are presented in Figures 4.3, 4.4, and 4.5. Figure 4.3 shows filled Mach contours of constant scale for angles of attack of  $0^\circ$  and  $20^\circ$ , both at  $M = 1.96$ . At  $\alpha = 0^\circ$ , the flow is very symmetric around the missile and a



(a)  $M = 1.96$ ,  $\alpha = 0^\circ$

(b)  $M = 1.96$ ,  $\alpha = 20^\circ$

Figure 4.3: Filled Mach contours, constant z-plane of symmetry.

symmetric oblique shock cone begins at the nose. The shock angle is approximately  $33^\circ$ , which matches closely with analytical data [3]. At  $\alpha = 20^\circ$ , an asymmetric shock comes off the bottom of the nose. Figure 4.3 demonstrates this change in the flow solution.

The wake is also greatly affected by the angle of attack of the missile, as is the pressure distribution over the body of the missile. Figure 4.4 shows the static pressure behind and on the surface of the missile. At an angle of attack of  $0^\circ$ , the static pressure on the surface of and behind the missile is very symmetric, as expected because of the missile symmetry. At  $\alpha = 20^\circ$ , however, large pressure gradients are visible both in the wake and on the missile body. These asymmetries in the static pressure on the surface of the missile are the cause of the restoring moment experienced by the missile.

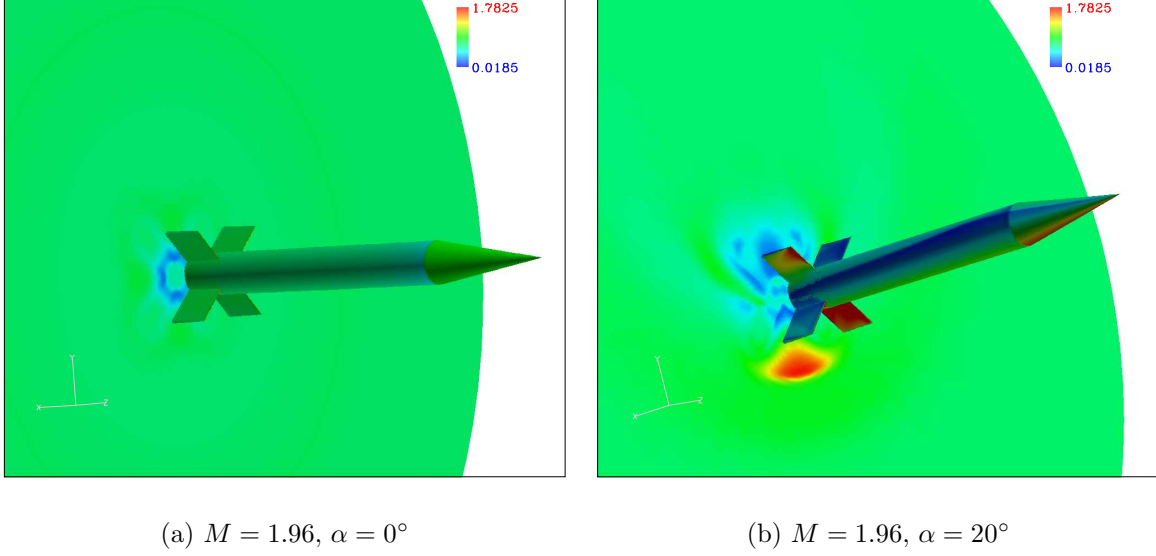
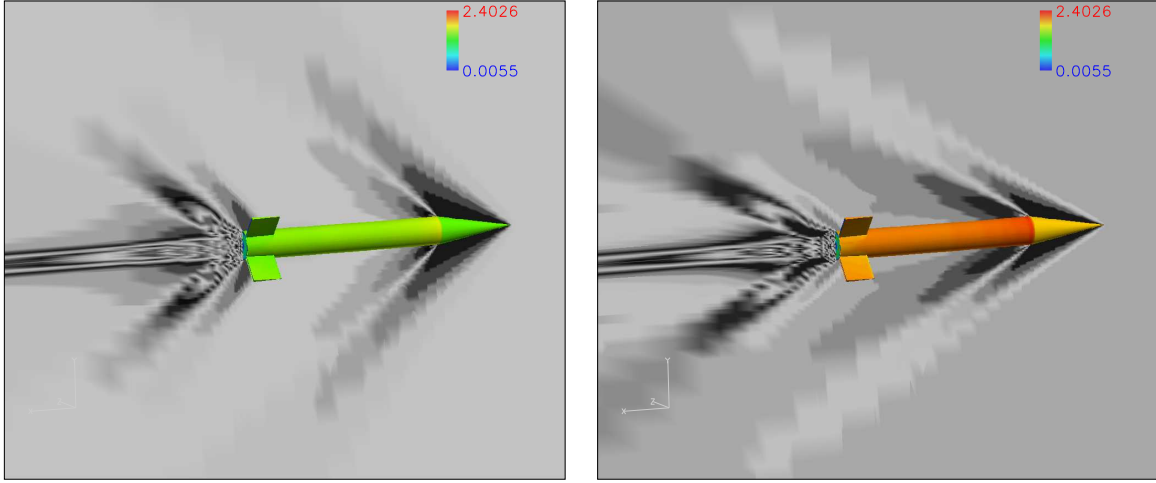


Figure 4.4: Static pressure on the missile surface and a constant x-plane in the wake.

Figure 4.5 shows zebra plots (alternating black and white) of Mach number with constant scale for Mach numbers 1.58 and 2.10, both at an angle of attack of  $0^\circ$ . The greater speed of the  $M = 2.10$  case caused a stronger, narrower oblique shock cone to develop off the nose and off the fins. Off the nose, the angles of the shock cones were found to be approximately  $31^\circ$  for  $M = 2.10$  and  $41^\circ$  for  $M = 1.58$ . Both values matched closely with analytical data, which predicted cone angles of approximately  $29^\circ$  and  $40^\circ$ , respectively [3].

Aside from using the static cases to initialize the dynamic cases, the cases run at  $M = 1.96$  with varying  $\alpha$  were also used to estimate the static stability coefficient,  $C_{m_\alpha}$ . The static stability coefficient, as described previously, is the slope of the  $C_m$  vs  $\alpha$  curve. Figure 4.6 shows this curve for the inviscid case along with comparison data from [16] and [33]. The measured moments match well with the comparison CFD data [16], which used an inviscid solver, but significant differences were seen between current CFD data and wind tunnel data [33] at low and high angles of attack. At low angles of attack, this is due to sting interference in the wind tunnel testing. Sting effects were found to be significant for angles of attack up to  $7^\circ$  [33]. Beyond  $\alpha = 10^\circ$ ,



(a)  $M = 1.58$ ,  $\alpha = 0^\circ$

(b)  $M = 2.10$ ,  $\alpha = 0^\circ$

Figure 4.5: Zebra plots of Mach number for the constant z-plane of symmetry.

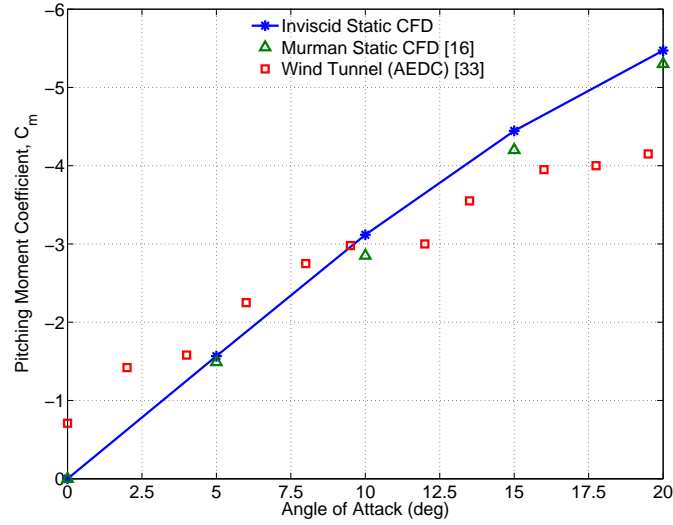


Figure 4.6: Inviscid pitching moment coefficient.

the inviscid CFD diverges again from the experimental data. The most likely reason is that the inviscid solver fails to model the flow separation that actually occurs at these angles of attack.

*4.1.2 Viscous.* Five viscous cases were run statically. These five shared a common Mach number of 1.96 and the angle of attack was varied from 0 to 20 degrees. Although a Roe-based solver was used for the final cases, stability issues required that each of the five cases be run first with a Steger-Warming solver. The Steger-Warming method is more dissipative than the Roe method and is not as difficult to get started. The Roe solver ran without issues for each case when it was initialized from the result of 1500 iterations with the Steger-Warming solver.

Although stability issues were overcome by initializing from a Steger-Warming solution, both methods had difficulties with residual convergence. As Figure 4.7 shows, neither the Steger-Warming nor the Roe methods converged to the desired three orders of magnitude. The case presented here,  $\alpha = 10^\circ$ , barely converged one order of magnitude using the Steger-Warming solver, and the residuals stalled before converging a single order of magnitude for the Roe solver. Even running an additional 1,000 iterations, none of the cases using the Roe method converged by more than one order of magnitude.

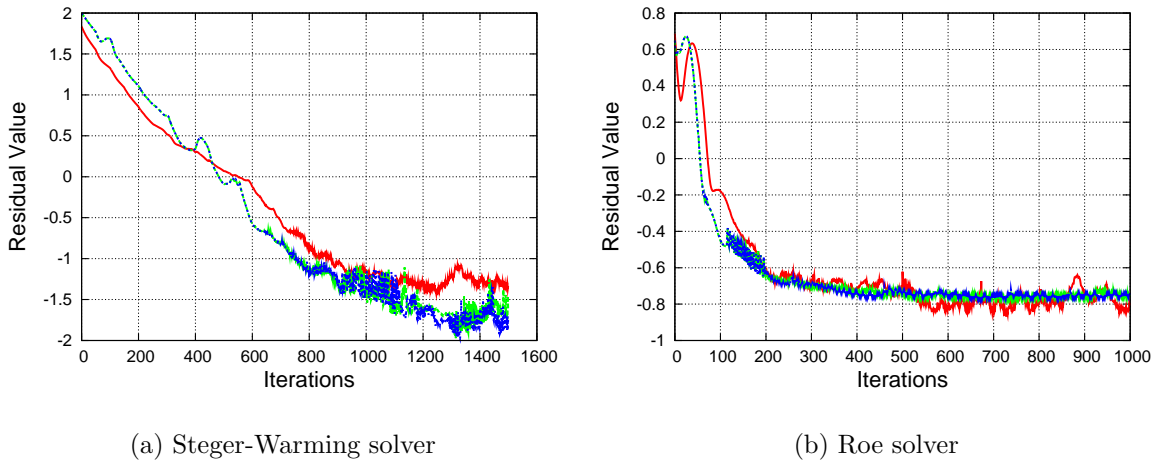


Figure 4.7: Viscous residual convergence.  $M = 1.96$ ,  $\alpha = 10^\circ$



Since the residuals did not converge to the desired degree, the force and moment coefficient histories were used to determine solution convergence. Figure 4.8(a) shows the transient part of the solution with the Steger-Warming solver. By the time the Steger-Warming solver finished and the Roe solver started (Figure 4.8(b)), the coefficients were basically converged, and only small changes were made. For this reason, these solutions were considered sufficiently converged to use for the present analysis.

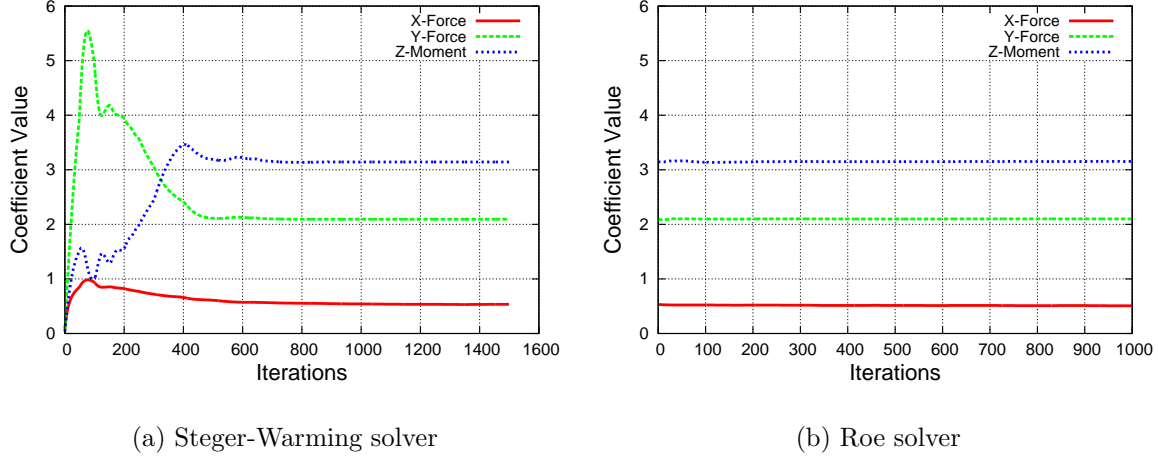


Figure 4.8: Viscous force and moment histories.  $M = 1.96$ ,  $\alpha = 10^\circ$

One major difference between the viscous and inviscid solvers is the way that boundary conditions are applied. For viscous solutions, the velocity at a surface is constrained to be zero, while an inviscid boundary merely requires that the component of velocity normal to the surface be zero. Figure 4.9 shows the surface velocity magnitude and contours of Mach number for a viscous and an inviscid case. Note that the velocity magnitude on the surface of the viscous case is a constant zero, while the flow is moving at varying speeds on the inviscid surface. The legend in Figure 4.9 refers to the contours of Mach number, which clearly show the shocks and expansions in the flow. The viscous and the inviscid solvers computed very similar solutions for the contours of Mach number and matched particularly well for the shock off the lower surface of the nose and the expansion above the nose. The main reason for the minor differences visible in the two solutions is that the viscous grid was much more refined.

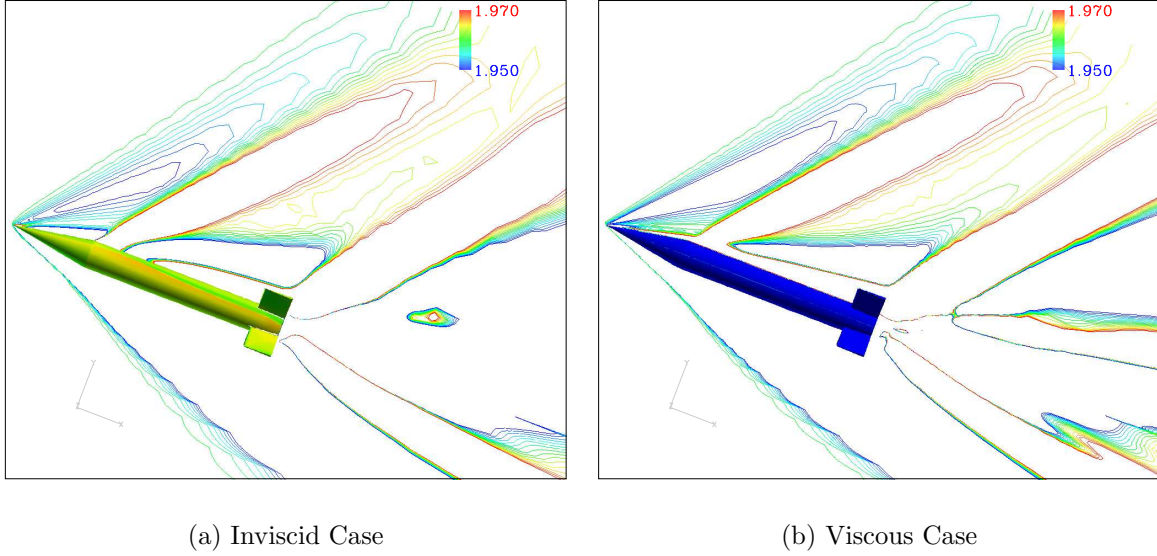


Figure 4.9: Comparison of surface velocity and Mach contours for viscous and inviscid cases.  $M = 1.96$ ,  $\alpha = 20^\circ$

Another comparison of the viscous and inviscid cases is shown in Figure 4.10. This case shows lines of surface flow on the missile body and a coordinate surface of constant x-location just in front of the fins shaded by stagnation pressure. The two images are very similar, but differences may be seen between the viscous and inviscid cases. Separation is visible in the lines of surface flow as lines join together and leave the surface. The viscous case displays a line of separation starting just after the nose and running along the side of the missile all the way to the tail. The inviscid case appears to exhibit some separation, but only starting at the rear of the missile. The coordinate surface of stagnation pressure shows the vortices developed by the separation over the body. Both cases show the development of twin vortices, but the vortices are much more clearly defined for the viscous case. This difference in definition is due to the fact that the viscous solver models the true physics of separated flow while the inviscid solver does not.

The primary goal of the present research, however, is not necessarily to resolve the flow around the missile as accurately as possible, but to predict the sum of the forces and moments about the missile center of gravity. Figure 4.11 shows that, for

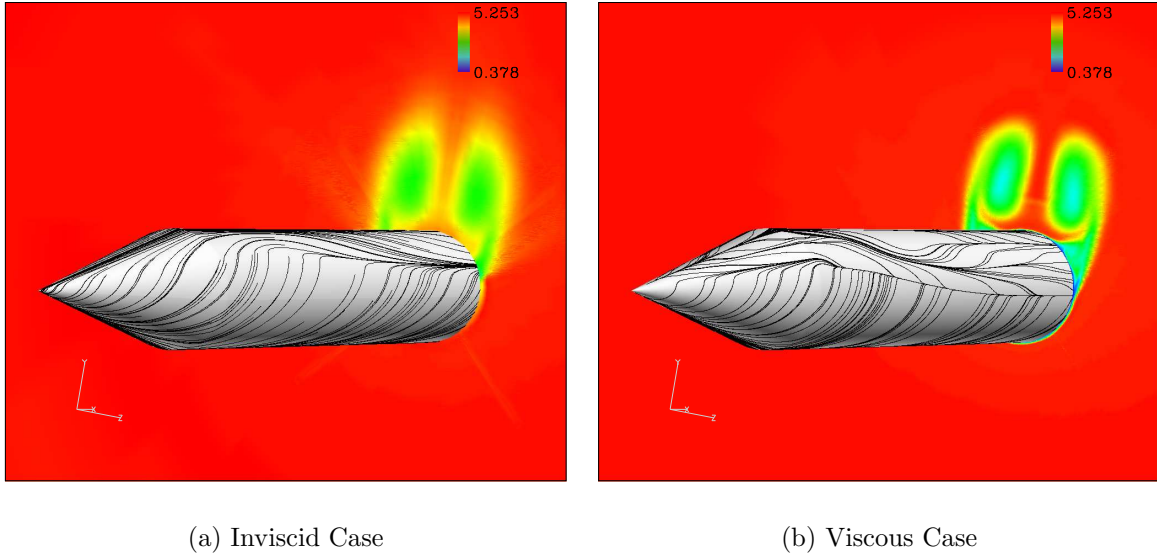


Figure 4.10: Comparison of surface flow and stagnation pressure for viscous and inviscid cases.  $M = 1.96$ ,  $\alpha = 20^\circ$

the static cases at least, there is good agreement between the viscous and inviscid methods for resolving the forces and moments on the missile. The solutions were expected to be similar, but the degree of agreement between them was not expected, especially at higher angles of attack. As Figure 4.10 shows, the viscous surface caused large vortices to form, and these vortices flowed directly over the upper tail surfaces, which was expected to change the forces and moments computed there. The inviscid solver also found some degree of separation, however, which led to very similar results. Another possible reason for the lack of difference is the sharp leading edge of the fins. This causes them to behave almost as ideal flat plates, for which the surface pressure is nearly unchanged with viscosity.

Certain other factors also may have affected the static viscous solutions. First of all, the Baldwin-Lomax turbulence model that was used was not designed for separated flows. Another turbulence model may have more accurately modeled the flow. Additionally, although the forces and moments for the static cases had converged to an acceptable degree, the residuals had not. Additional timestep ramping and additional iterations may have caused the residuals to converge further, and may

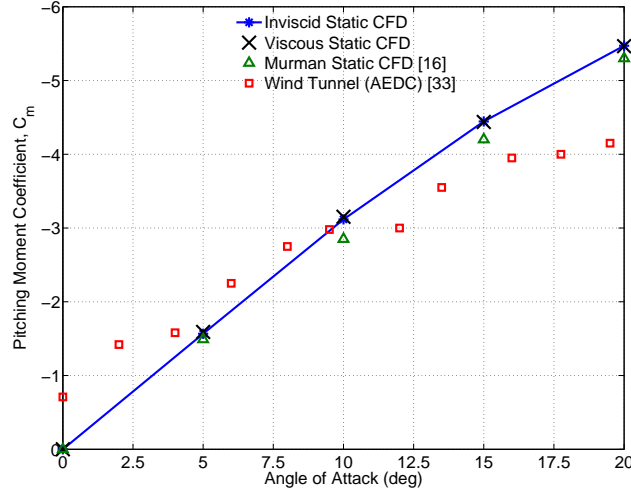


Figure 4.11: Viscous and inviscid pitching moment coefficient.

have improved solution accuracy. Also, to ensure that correct solutions were obtained by the viscous solver, a grid convergence study would need to be undertaken. This process would involve refining the grid until changes in the solution were no longer observed with an increase in the number of cells.

## 4.2 Prescribed Motion Parameter Selection

*4.2.1 Pitch: Forced Oscillation.* As described in Chapter III, certain parameters defining the sinusoidal oscillation were chosen with care, since they had the potential to affect the solution. These parameters included the reduced pitch rate, the amplitude of oscillation, the number of iterations per oscillation, the number of Newton iterations, and the total number of oscillations.

The forced oscillation is defined by Equation 3.1, which is repeated below for convenience:

$$\alpha = \alpha_o + \alpha_m \sin(\omega t) \quad (4.1)$$

As described previously, this motion is fully defined by the choices of reduced pitch rate ( $k_q$ ) and oscillation amplitude ( $\alpha_m$ ). Initial estimates for each of these values were gained from comparison data, and they were varied together and separately to determine their effect upon the static and dynamic stability coefficients. Many levels of testing were performed before each of the parameters were selected. Initially many amplitudes were tested with many reduced pitch rates in order to find appropriate values to use in subsequent tests. Presented here is the final level of the testing: only one parameter varied at a time, except for the number of iterations per oscillation, which was varied with nearly every case to ensure timestep convergence.

All parameter convergence tests were run using the  $M = 1.96$  and  $\alpha = 5^\circ$  case. Other cases were tested to a lesser extent, but this case was chosen as the characteristic case for all angles of attack and Mach number.

*4.2.1.1 Reduced Pitch Rate.* Twelve reduced pitch rates were tested to determine the optimal value. The number of iterations per oscillation was varied concurrently with the pitch rate so that the effects of time discretization at each pitch rate could be observed.

Increasing the pitch rate increased the area of the  $C_m$  vs  $\alpha$  curve, as shown in Figure 4.12. To explain this increase in area, consider again Figure 2.4, which illustrated the induced velocity and change in effective angle of attack on the missile tail due to pitching rate. The induced velocity is equal to  $ql_t$  and the induced angle of attack is  $\Delta\alpha = \frac{ql_t}{V_\infty}$ , so, the induced angle of attack increases linearly with the pitch rate. This in turn induces a change in the lift of the tail and a change in moment at the center of gravity opposing the rotation. Figure 4.12 confirms this expectation, showing greater deviations from the static moment coefficient with greater pitch rates.

Note also that the value of  $C_m$  at the angle extrema appears to be constant with the changing pitch rate, which means that the slope between the extrema,  $C_{m\alpha}$  is relatively constant as well. In Figure 4.12, the pitch damping for each case is found from the difference in  $C_m$  as the model pitched up and down through  $\alpha = 5^\circ$ .

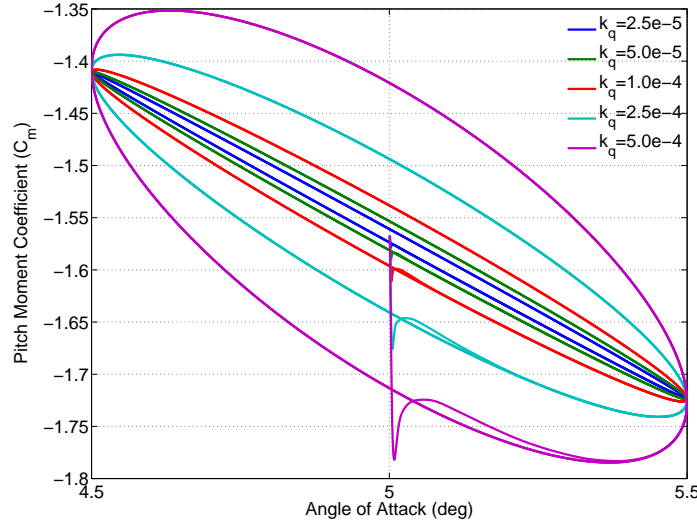


Figure 4.12:  $C_m$  vs  $\alpha$  histories with varying pitch rate.  $M = 1.96$ ,  $\alpha = 5^\circ$

Figure 4.13 shows the angle, lift coefficient, and moment coefficient histories plotted against iterations for three reduced pitch rates. For a series of static solutions, the peaks of angle and lift coefficient would line up identically with each other and with the troughs of the pitch moment coefficient curve. Notice the dotted vertical line on all three graphs lining up with the third peak in angle of attack. Observe that the peaks of the force and moment coefficients for the slowest pitch rate,  $k_q = 0.00025$ , lead the peaks of the angle of attack by just a small amount. As the pitch rate increases to  $k_q = 0.001$ , the amount of lead also increases. This lead in the system is a product of the hysteresis effects of the dynamic system, again caused by the fact that the increasing pitch rate leads to an increasing induced velocity on the tail. In this example, maximum induced velocities occur as the missile passes through an angle of attack of  $5^\circ$ . As the missile pitches more rapidly, the extrema of the  $C_m$  curve move toward the location of maximum induced velocity.

As described in Chapter III, the pitch damping is found as the slope of a  $C_m$  vs  $k_q$  curve. If the damping was completely independent of the pitch rate, then the  $C_m$  vs  $k_q$  plot would be a straight line. As shown in Figure 4.14(a), however, the slope of the  $C_m$  vs  $k_q$  curve is not a constant, meaning that different rates result in

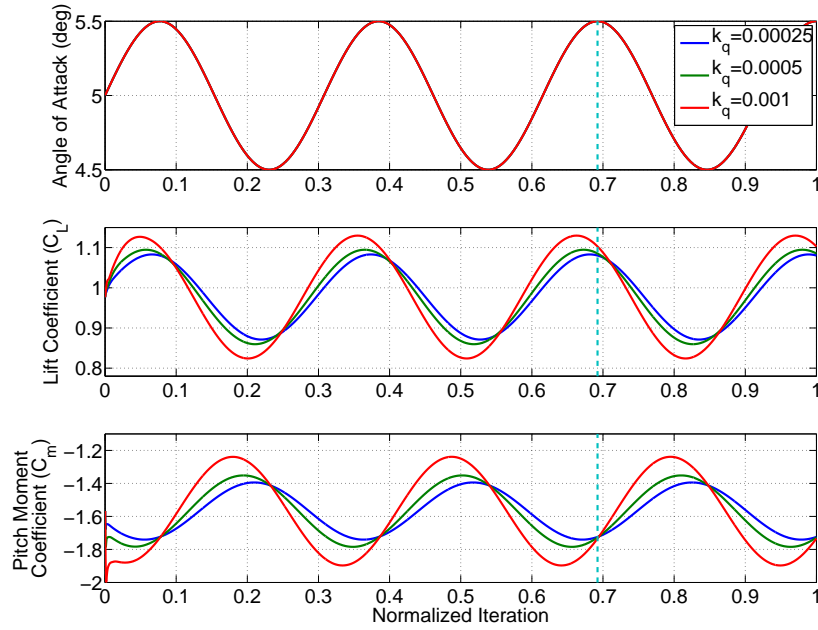
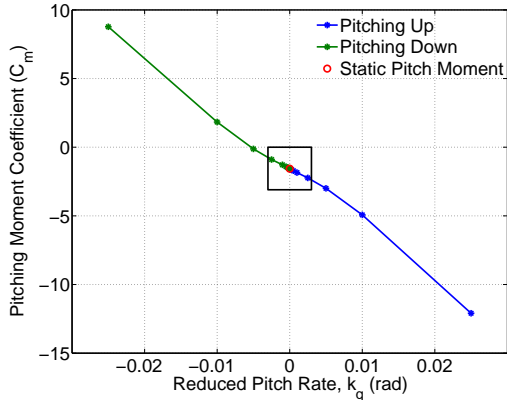


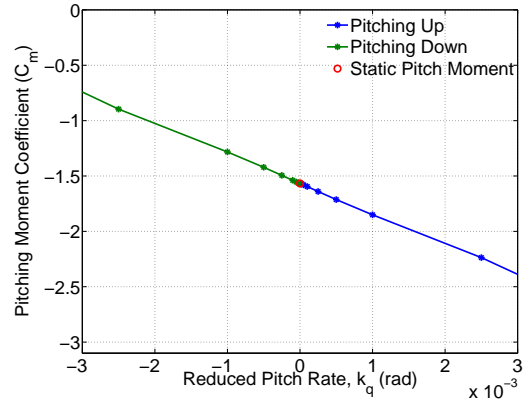
Figure 4.13: Angle of attack, lift coefficient, and pitching moment coefficient histories.  $M = 1.96$ ,  $\alpha = 5^\circ$

different damping coefficients. Figure 4.14(b) shows a zoomed in view, for which more of a linear relation is observed. This shows that the damping coefficient is relatively constant with low reduced pitch rate, but the induced velocities caused by higher reduced pitch rates result in nonlinear behavior.

The effect of reduced pitch rate on the static and dynamic stability coefficients for cases with 1600 iterations per oscillation may be seen in Table 4.1. The static and dynamic coefficients were found to be relatively constant in the range of  $k_q = 0.0001$  to  $k_q = 0.001$ , meaning that the solution is relatively independent of the rate within that range. Figure 4.15 demonstrates these effects graphically for 1600 and 3200 iterations per oscillation. Note that both of the stability coefficients calculated were relatively constant with the doubling of the number of iterations per oscillation over the limited range of  $k_q$  mentioned above, but large variations were seen outside of that range.



(a) High pitch rates lead to nonlinear  $C_m$  vs  $\alpha$  relation.



(b) Linear  $C_m$  vs  $k_q$  relation is observed at lower pitch rates.

Figure 4.14: Pitch damping coefficient is the slope of  $C_m$  vs  $k_q$ .  $M = 1.96$ ,  $\alpha = 5^\circ$

Table 4.1: Effect of reduced pitch rate on stability coefficients:  $1600 \frac{\text{iterations}}{\text{oscillation}}$ .

$k_q$	$C_{m_\alpha}$	% Change	$(C_{m_q} + C_{m_{\dot{\alpha}}})$	% Change
0.000010	-17.8929		-102.0000	
0.000025	-17.8935	0.00%	-218.6000	114.31%
0.000050	-17.8963	0.02%	-258.3000	18.16%
0.000100	-17.9089	0.07%	-279.5000	8.21%
0.000250	-18.0081	0.55%	-291.1000	4.15%
0.000500	-18.3180	1.72%	-291.5700	0.16%
0.001000	-19.0520	4.01%	-283.9050	2.63%
0.002500	-13.7183	28.00%	-268.0820	5.57%
0.005000	9.1347	147.95%	-286.9930	7.05%
0.010000	48.8813	456.32%	-338.2760	26.18%
0.025000	-10.1992	120.87%	-417.3322	23.37%
0.050000	-659.5501	6366.68%	-411.6254	1.37%



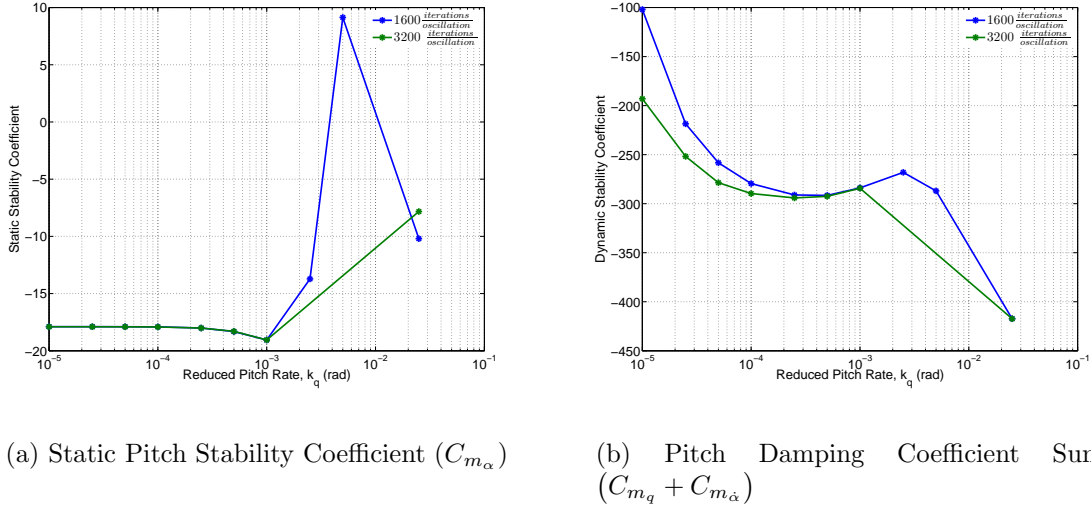


Figure 4.15: Pitch stability coefficients as a function of  $k_p$  for two values of iterations per oscillation.  $M = 1.96$ ,  $\alpha = 5^\circ$

At low pitch rates, this variation was most likely due to a lack of full convergence; more iterations per oscillation were required. A more in-depth timestep convergence study was performed on several pitch rates to investigate this. For the case with  $k_q = 0.000025$ , doubling the number of iterations from 1600 to 3200 per oscillation changed the pitch damping term in excess of 15%. This implies that additional timesteps would be required to fully resolve this solution. For the case with  $k_q = 0.00025$ , on the other hand, the same doubling of iterations per oscillation changed the solution by only 1.04%. Cases with slower pitch rates require a larger number of iterations per oscillation because the slower rate of oscillation means that additional physical time elapses during the same pitching cycle. In order to have the same number of iterations per oscillation as a case that is pitching more rapidly, a larger timestep is required, which, in turn has the potential to degrade the flow solution, as seen here.

The discrepancies seen in the static and dynamic coefficients seen at higher pitch rates are again due to the nonlinearities introduced by the large induced velocities on the tail. A timestep convergence study on the case with  $k_q = .025$  showed that the damping coefficient was virtually constant (and larger than predicted at lower rates) for a large range of iterations per oscillation. The static stability coefficient, on the

other hand, was still changing by a margin of over 20% as the number of iterations was increased from 1600 to 3200 per oscillation, implying that additional iterations per oscillation were required to achieve convergence. Based upon this analysis the reduced pitch rate chosen for the final forced oscillation testing was  $k_q = .00025$ .

*4.2.1.2 Oscillation Amplitude.* Various oscillation amplitudes were tested in conjunction with the number of iterations per oscillation. Figure 4.16 shows the effect of changing amplitude on the  $C_m$  vs  $\alpha$  cycle for the test case with  $M = 1.96$ ,  $\alpha_o = 5^\circ$ . The moment coefficient as the model passed through  $\alpha = 5^\circ$  appears

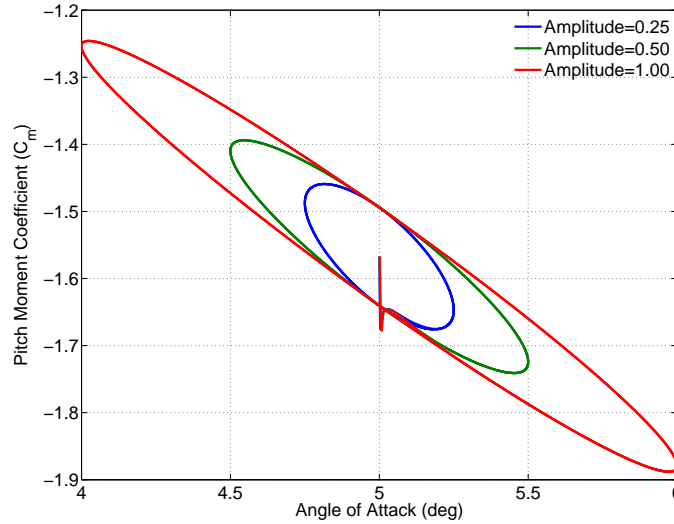
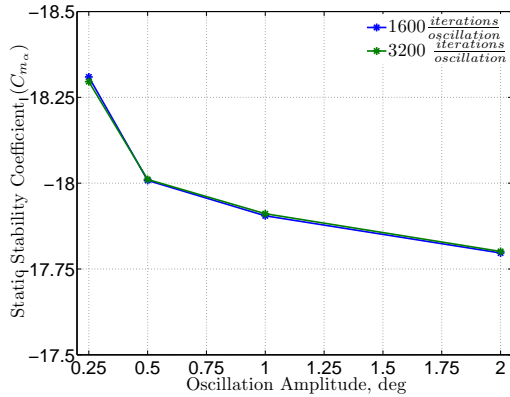


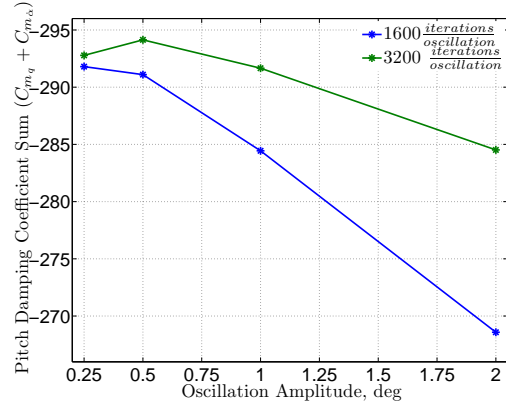
Figure 4.16:  $C_m$  vs  $\alpha$  for multiple amplitudes.  $M = 1.96$ ,  $\alpha_o = 5^\circ$

constant with changing amplitude, as did the  $C_m$  vs  $\alpha$  slope for the angle extrema. Even small variations in the moments calculated, however, lead to noticeable changes in the stability coefficients, as shown in Figure 4.17.

Figure 4.17 shows the effect of both amplitude and iterations per oscillation on the stability coefficients as the number of iterations was increased from 1600 to 3200 per oscillation. With any amplitude, only minute changes were seen in the static stability coefficient as the number of iterations was increased, but larger changes were observed in the damping coefficient. Especially with larger amplitudes, the



(a) Static Pitch Stability Coefficient ( $C_{m_{\alpha}}$ )



(b) Pitch Damping Coefficient Sum ( $C_{m_q} + C_{m_{\dot{\alpha}}}$ )

Figure 4.17: Pitch stability coefficients as a function of amplitude for two values of iterations per oscillation.  $M = 1.96$ ,  $\alpha = 5^\circ$

predicted damping coefficients changed to a large degree with the number of iterations per oscillation. At  $\alpha_m = 2^\circ$ , doubling the number of iterations per oscillation from 1600 to 3200 increased magnitude of the predicted damping coefficient by nearly 6%, implying that another increase in the number of iterations would increase magnitude of the solution by a still significant degree. With  $\alpha_m = 0.5^\circ$ , on the other hand, the predicted damping value changed by only 1.04% with the same increase in iterations per oscillation. This shows that the solution had converged for the  $\alpha_m = 0.5^\circ$  case with only 1600 iterations. Additional testing showed that further increasing the number of iterations per oscillation for the  $\alpha_m = 1^\circ$  and  $\alpha_m = 2^\circ$  caused these cases to converge to the same damping coefficient as the  $\alpha_m = 0.5^\circ$  case. For computational efficiency,  $\alpha_m = 0.5^\circ$  was chosen for the final test cases.

#### 4.2.1.3 Newton Iterations.

The number of Newton iterations was important because time-accurate flow solving requires the convergence of these inner iterations. As Figure 4.18 shows, the number of Newton iterations was found to have only a small effect on the calculation of stability coefficients for the case with  $\alpha = 5^\circ$ ,  $M = 1.96$ ,  $\alpha_m = 0.5$ , and 1600 iterations per oscillation. The static stability

coefficient changed by only 0.13% as the number of Newton iterations was increased from four to eight, and the pitch damping coefficient changed by only 0.72%. For this reason, the number of Newton iterations was set to four for all of the dynamic tests.

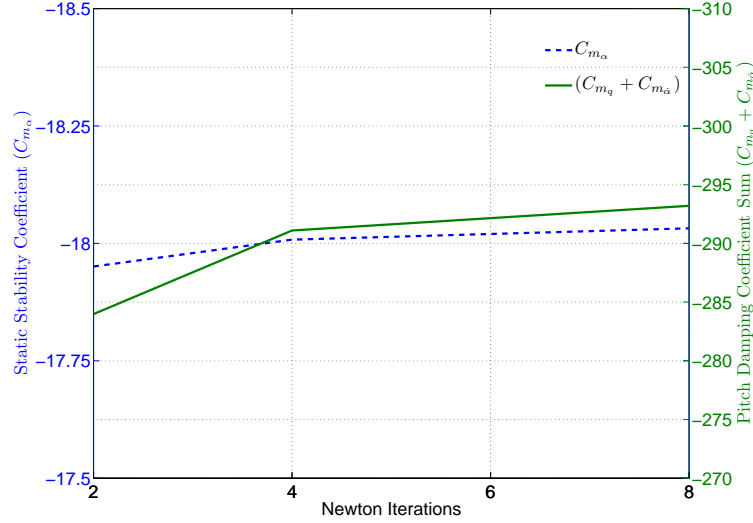


Figure 4.18: Effect of Newton iterations on pitch stability coefficients.  $M = 1.96$ ,  $\alpha = 5^\circ$

**4.2.1.4 Oscillation Number.** To test the appropriate number of oscillations, one case was run for four complete oscillations. Figure 4.19(a) shows the pitching cycle vs pitching moment coefficient. Figure 4.19(b) zooms in on the lower portion of this plot to show that, after the initial transients die out, each successive cycle follows the same path. Although the solution was converged by the second oscillation for this test case, all cases were run for 3 full oscillations to ensure that the periodic moment coefficients were properly resolved.

**4.2.1.5 Iterations per Oscillation.** The convergence study of iterations per oscillation was performed concurrently with the convergence studies of all other parameters. Based upon the above analysis, the number of iterations per oscillation was set to 1600. This number achieved the desired convergence for the parameters chosen above, while at the same time minimizing the computational expense.

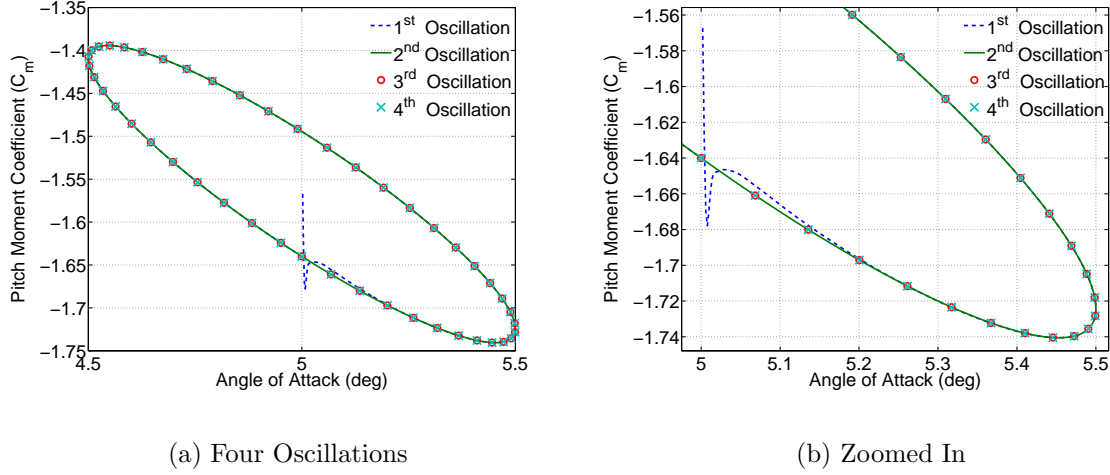


Figure 4.19: Multiple cycles show oscillation convergence.  $M = 1.96$ ,  $\alpha = 5^\circ$

**4.2.2 Roll: Constant Revolution.** The roll motion of the missile was defined solely by the reduced roll rate,  $k_p = pd/2V_\infty$ . As described in Chapter III, the reduced roll rate and the number of iterations per revolution were varied to determine their effect upon the solution and to guarantee solution convergence. Initial estimates for  $k_p$  were found from Oktay and Akay [21], where  $k_p = 0.00326$  was used. Figure 4.20 displays the change in  $C_l$  as a function of reduced roll rate with the number of iterations per revolution held constant at 23,040. Note the nearly constant slope, which confirms that the roll damping is nearly independent of the reduced roll rate.

The roll damping was not found to be entirely independent of roll rate, however, as shown in Figure 4.21. This figure shows the roll damping coefficient as a function of the reduced roll rate for two numbers of iterations per oscillation. Note that, for this case, the slower rates of rotation appear to be changing to a large degree with the number of iterations, while the faster reduced roll rates remain nearly constant. Additional iterations may have caused the slowest reduced roll rates to converge to the same value as the faster rates, but the computational expense was considered unnecessary since the faster rates converged sufficiently. Based on these tests,  $k_p = 0.0025$  was chosen as the the reduced roll rate for the six final test cases.

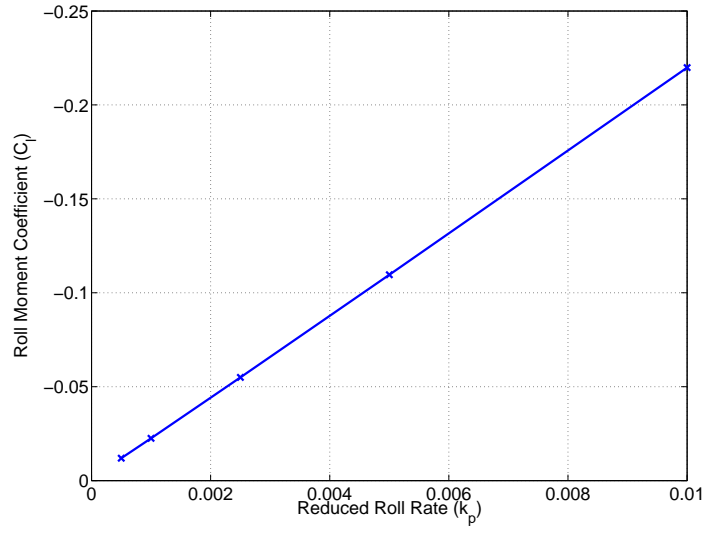


Figure 4.20: Roll moment coefficient for differing rates.  $M = 1.96$ ,  $\alpha = 5^\circ$

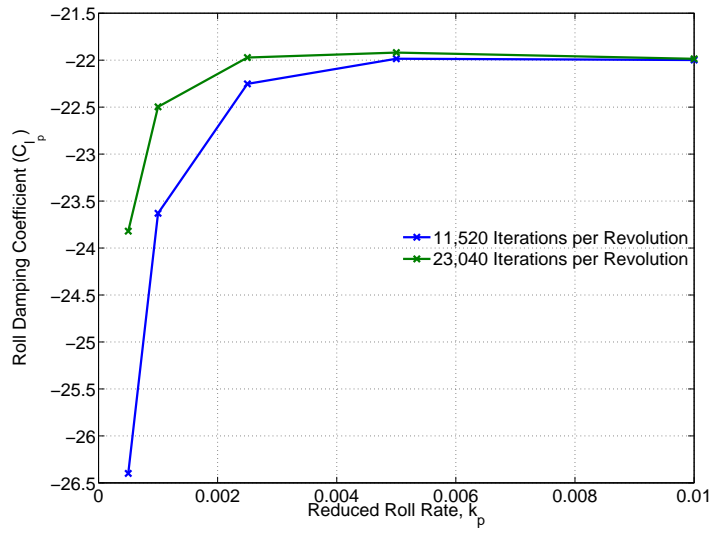


Figure 4.21: Roll damping coefficient as a function of rate and number of iterations.  $M = 1.96$ ,  $\alpha = 5^\circ$

Table 4.2 highlights the effect of iteration per revolution with reduced roll rate equal to 0.0025. As the number of iterations per revolution was doubled, the solution continued to change, but with diminishing returns. Doubling the number of iterations per revolution from 23,040 to 46,080 changed the roll moment coefficient by only 0.18%, and  $C_{l_p}$  by only 0.32%. Since this change was so small, the number of iterations per revolution chosen for the six final test cases was 23,040.

Table 4.2: Effect of iterations per revolution on roll moment and damping,  $k_p = 0.0025$ .

<u>Iterations Revolution</u>	$C_l$	% Change	$C_{l_p}$	% Change
360	-0.1126		-45.0560	
720	-0.0944	16.16%	-37.7610	16.19%
1440	-0.0737	21.93%	-29.5000	21.88%
2880	-0.0630	14.52%	-25.1890	14.61%
5760	-0.0578	8.25%	-23.1310	8.17%
11520	-0.0556	3.81%	-22.2520	3.80%
23040	-0.0549	1.26%	-21.9720	1.26%
46080	-0.0548	0.18%	-21.9010	0.32%

*4.2.3 Test Parameter Recap.* For convenience, the parameters used for all of the forced motion cases are given here. The parameters chosen for forced pitch oscillation were:

$$\begin{aligned}
\text{Iterations per Oscillation} &= 1600 \\
\text{Oscillations per Test} &= 3 \\
\text{Newton Iterations} &= 4 \\
\alpha_m &= 0.5^\circ \\
k_q &= 0.00025
\end{aligned} \tag{4.2}$$

For rolling motion, the parameters chosen were:

$$\begin{aligned} \text{Iterations per Revolution} &= 23040 \\ k_p &= 0.0025 \end{aligned} \tag{4.3}$$

### 4.3 Pitch Stability Derivatives: $\alpha = 0 - 20^\circ$

Five of the inviscid and all of the viscous cases were run at a constant Mach number, angle of attack varying from  $0 - 20^\circ$ . These cases were used to determine the stability coefficients as a function of angle of attack. Figure 4.22 shows the  $C_m$  vs  $\alpha$  loops for two inviscid cases, both with Mach number equal to 1.96. Mean angles of attack were 10 and 15 degrees.

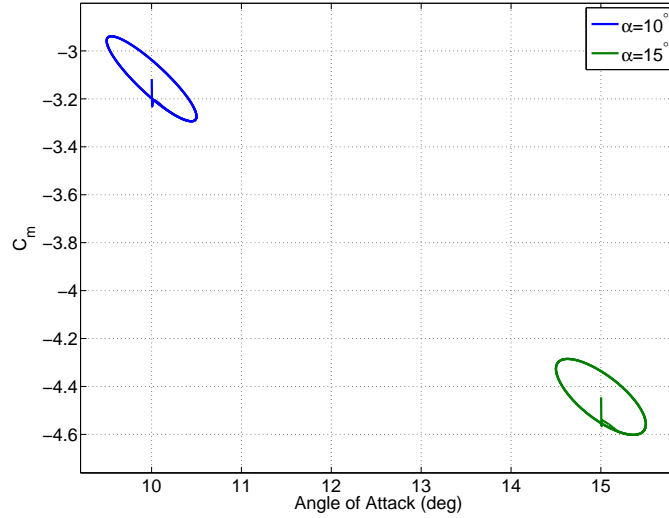


Figure 4.22: Pitching moment cycle for two angles of attack.  $k_q = 2.5e - 4$

Note the increased width of the case run at  $15^\circ$ . Similar to Figure 4.12, this increased width implies larger induced velocities and thus larger changes in the pitching moment about the center of gravity. Unlike Figure 4.12, however, which had the increased difference in  $C_m$  at the mean angle normalized by an increasing pitch rate, the cases in Figure 4.22 were oscillated at the same reduced pitch rate, so the larger



area of the pitch cycle implies a larger predicted damping. This is investigated further below.

*4.3.1 Static Pitch Stability.* Four computational sources were used for calculating the static stability coefficient,  $C_{m_\alpha}$ , as a function of angle of attack for  $M = 1.96$ . These four sources were inviscid static solutions, inviscid forced oscillation, viscous forced oscillation, and inviscid free oscillation. The data reduction techniques for each of these sources are given in Chapter III, and Figure 4.23 shows a comparison of the results from each method.

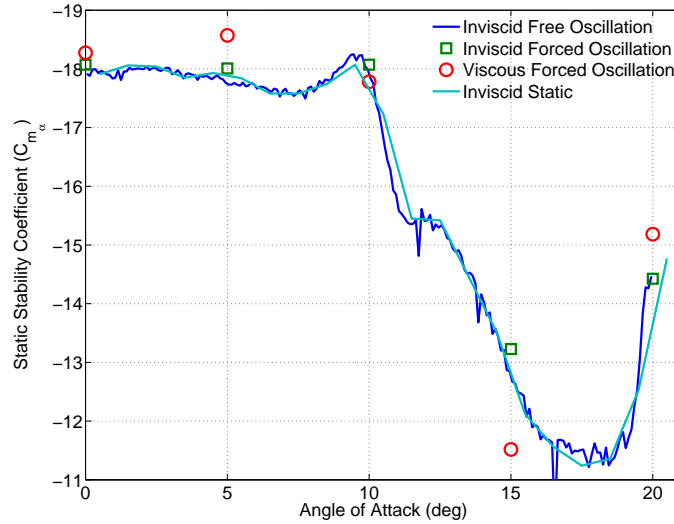


Figure 4.23: Comparison of static stability coefficients,  $M = 1.96$ .

In order to refine the stability coefficient calculated from the static solutions, additional static solutions were run at each angle of attack in the range  $1 - 21^\circ$ . Figure 4.23 shows that all four of the testing methods computed strikingly similar curves for  $C_{m_\alpha}$ . This result was expected for the three inviscid cases, since the oscillation rate of the forced oscillation was chosen carefully to avoid nonlinearities and the static and free oscillation solutions were based solely upon the model and flow properties.

The agreement between the viscous solver and the inviscid techniques was expected at low angles of attack because, in general, inviscid solvers compute accurate

surface pressures for the attached flows seen at low angles. At angles of attack of  $15^\circ$  and  $20^\circ$ , however, separated flow and large vortices off the body were expected, and both were observed in the viscous solution, as shown in Figure 4.10. As discussed above, however, a certain degree of non-physical inviscid separation was also modeled, which may have lead to some similarity between the solutions.

In addition to the possible reasons previously given for error in the static viscous solutions, the dynamic cases may have introduced their own error. A plot of the forced oscillation pitching cycle for a viscous case showed that the solution had not sufficiently converged to a consistently repeating cycle after the three oscillations that were run for each case. This failure to achieve cyclical convergence could introduce error into the viscous solutions. Two choices may have fixed this issue: additional oscillations or a decreased timestep.

*4.3.2 Pitch Damping.* The pitch damping coefficients were also determined at each angle of attack using the three dynamic methods mentioned above. These values, displayed in Figure 4.24 showed good agreement with other CFD methods [16, 17] for all angles of attack. At low angles of attack, all results from Beggar matched well with ballistic range data [31], but not with wind tunnel data [33]. This error is due to the sting effects of the wind tunnel testing for angles of attack up to  $7^\circ$  [33]. At high angles of attack, the inviscid CFD methods over-predict the value of the damping coefficient when compared to wind tunnel data. This is likely due to non-linear effects in the inviscid solution at high angles of attack, and the failure of the inviscid solver to model separation effects.

The viscous solver, on the other hand, does model the flow separation that occurs at high angles of attack, but the computed damping values agreed quite closely with those found from inviscid techniques. The viscous damping does diverge from the inviscid solutions at  $\alpha = 20^\circ$  and come closer to the damping values computed experimentally, but it was expected that the results of the inviscid and viscous cases at high angles of attack would be fundamentally different and that the viscous solution

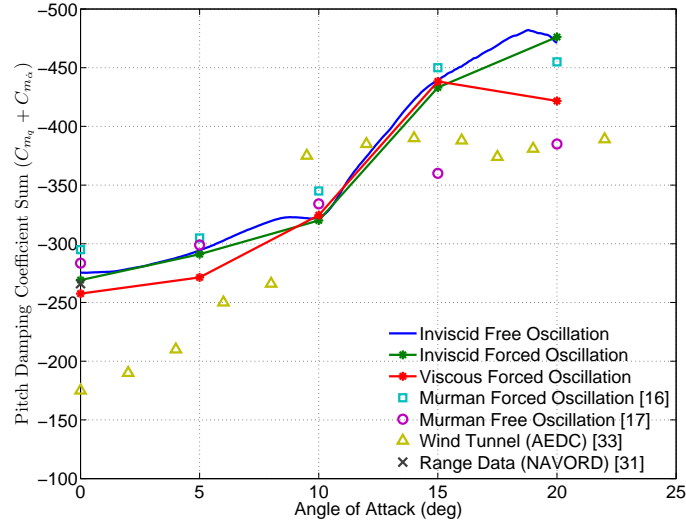


Figure 4.24: Comparison of dynamic stability coefficients,  $M = 1.96$ .

would more accurately model the experimental data. The possible explanations for this divergence of expectations and results for the pitch damping are the same as the reasons given above for the static pitch stability.

*4.3.3 Trajectory Prediction.* The equation of motion for a pure pitching system was developed in Chapter II and is repeated here in homogeneous form, since control surface deflections are not being analyzed.

$$\Delta\ddot{\alpha} - (M_q + M_{\dot{\alpha}})\Delta\dot{\alpha} - M_{\alpha}\Delta\alpha = 0 \quad (4.4)$$

The motion of the system is controlled by the static pitch stability derivative and the pitch damping derivative sum. Because both of these terms are functions of angle of attack, Equation 4.4 was integrated in time using linear interpolations between the discrete values found from the inviscid forced oscillation cases. The result of this integration for a case with an initial angle of attack of  $20^\circ$  is shown in Figure 4.25, along with the free oscillation result at the same initial angle and the same Mach number of 1.96.

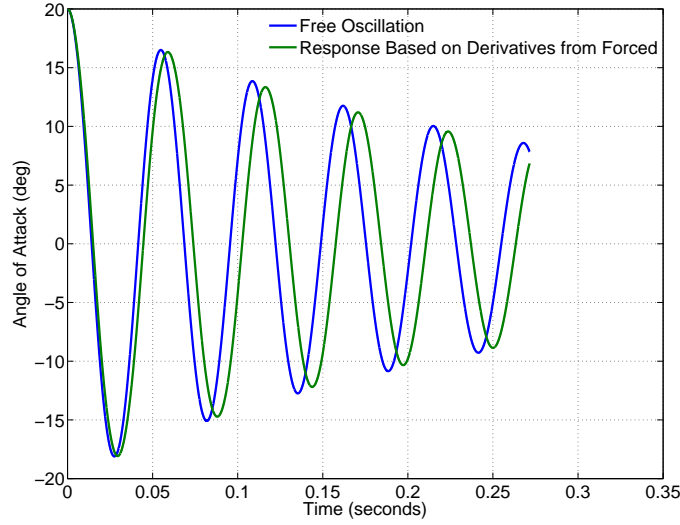


Figure 4.25: Comparison of free oscillation trajectory and trajectory predicted based on stability coefficients from forced oscillation.  $M = 1.96$ ,  $\alpha_{start} = 20^\circ$

The two methods match up well at the start of the trajectory, but the integration using values from forced oscillation shows a distinct and growing phase lead and slightly higher damping than the free oscillation case. These discrepancies are most likely due to the inaccuracies evident at high angles of attack, and, because the range of motion is quite large, even small degrees of error propagate through the solution to become large errors over time. Another possible source of error is that the second order model given in Equation 4.4 does not accurately model high amplitude motion, since it was based on a small perturbation assumption. Higher resolution might be possible with a higher order model that incorporates, for example, third order terms like  $C_{m_{\alpha\alpha}}$ .

Another case was examined to determine whether the stability derivative predicted by the forced oscillation would provide a more accurate modeling of the free oscillation at lower angles of attack. This case was started at an angle of attack of  $5^\circ$ , the trajectory of which is shown in Figure 4.26 along with the trajectory predicted by the second order model using forced oscillation stability derivatives.

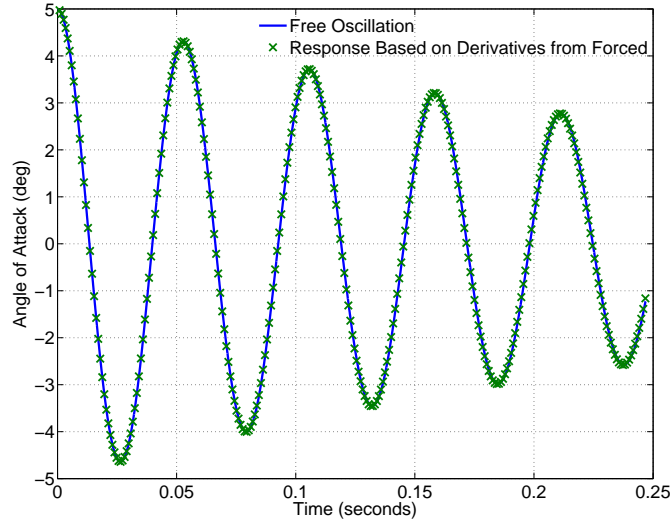
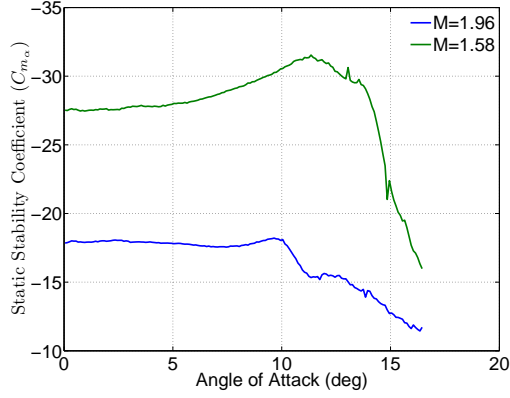


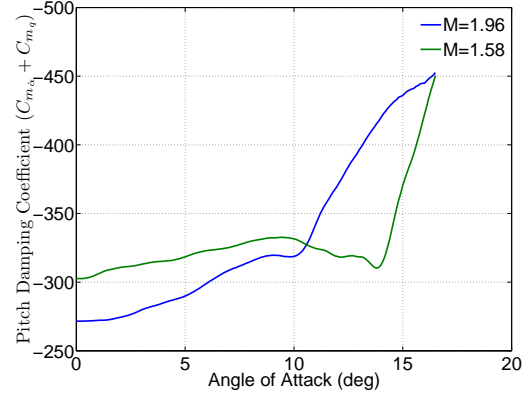
Figure 4.26: Comparison of free oscillation trajectory and trajectory predicted based on stability coefficients from forced oscillation.  $M = 1.96$ ,  $\alpha_{start} = 5^\circ$

Figure 4.26 shows that the stability derivatives estimated through forced oscillation, combined with second order model of the motion, predicted motion almost identical to the free oscillation case. After three full cycles, the equations of motion with coefficients from forced oscillation predicted a peak just 0.243% greater than the free free oscillation model, and exhibited lag of only 1.24 degrees. For the case in Figure 4.25, on the other hand, the third peak of the trajectory for the integrated method was 4.75% beneath the free oscillation trajectory, and lead by 60.3 degrees.

The final free oscillation case was run at  $M = 1.58$  and started from an angle of  $20^\circ$ . The two free oscillation cases initialized at  $\alpha = 20^\circ$  are shown together in Figure 4.27. Note that the static stability coefficient of the  $M = 1.58$  case is significantly higher at all angles of attack, and the values of the damping coefficient are higher for the lower Mach number until  $\alpha = 12^\circ$ , when the damping of the  $M = 1.96$  case spiked due to non-linearities. The  $M = 1.58$  case also experienced non-linearities at high angles of attack, but not until around  $\alpha = 14^\circ$ . The reasons for higher coefficient values at the lower Mach number is discussed in the following section.



(a) Static pitch stability coefficient.



(b) Pitch damping coefficient.

Figure 4.27: Static and dynamic stability for two Mach numbers.

#### 4.4 Pitch Stability Derivatives: $M = 1.58 - 2.50$

The remaining five inviscid forced oscillation tests were run with a constant angle of attack of zero degrees,  $M = 1.58 - 2.50$ . Figure 4.28 shows the pitch moment coefficient vs angle of attack for three Mach numbers. Note that the magnitude of

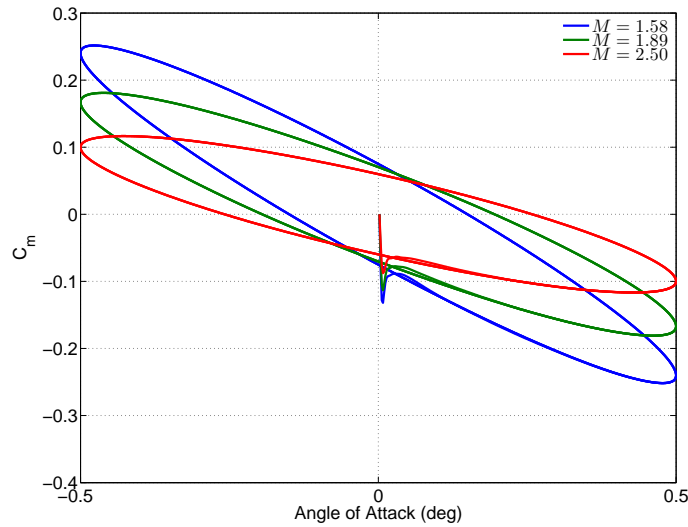


Figure 4.28:  $C_m$  vs  $\alpha$  cycles for multiple Mach numbers.

the slope between left and right sides ( $C_{m_{\alpha}}$ ) decreases with increasing Mach number. Additionally,  $\Delta C_m$  at  $\alpha = 0^\circ$  for the  $M = 2.50$  case may be seen to be smaller than

it is for the two slower cases. The implications of and reasons for these results are discussed below.

*4.4.1 Static Pitch Stability.* The decrease in  $C_{m_\alpha}$  with increasing Mach number is visible in Figure 4.28. It is even more evident in Figure 4.29, which shows excellent agreement with data from the Ballistic Research Laboratory (BRL). All Mach numbers tested agree very closely with the experimental range data. This

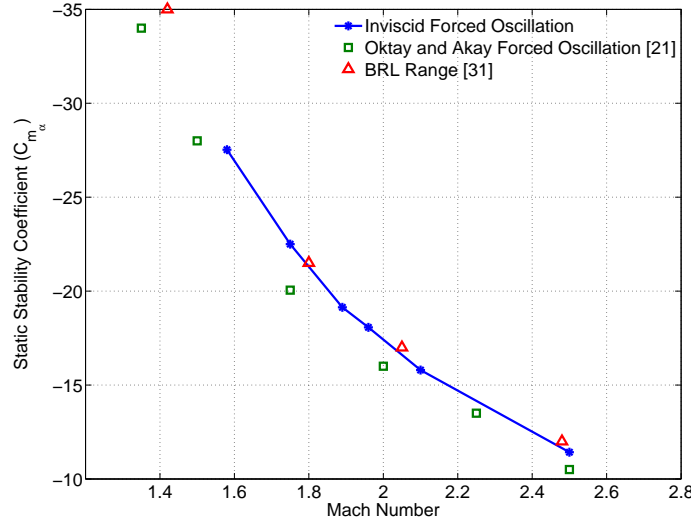


Figure 4.29: Static pitch stability as a function of Mach number.

trend in  $C_{m_\alpha}$  follows the trend predicted by linear wing theory and Newtonian impact theory applied at supersonic Mach numbers. These theories show that the normal force coefficient on a lifting surface decreases with increasing Mach number [11]. This in turn decreases the restoring moment provided by the tail, and since the moment provided by the tail is the main source of static stability for the Basic Finner, the overall static stability of the missile decreases as well.

*4.4.2 Pitch Damping.* Comparison damping coefficient data exhibited more scatter with varying Mach number, but Figure 4.30 shows that, in general, inviscid forced oscillation techniques using Beggar captured the damping coefficients accurately. The largest degree of variance is seen at  $M = 2.5$ .

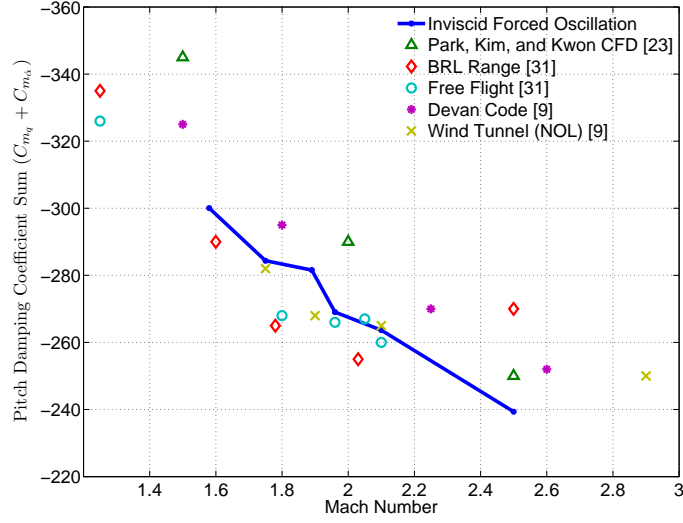


Figure 4.30: Pitch damping as a function of Mach number.

The general trend in the pitch damping coefficient is to decrease with increasing Mach number. This is the expected behavior, because the damping term is based on the induced velocity on the tail,  $ql_t$ . This term was held constant for all Mach numbers, but the induced angle of attack on the tail is  $ql_t/V_\infty$ . This means that the change in angle of attack induced by the pitch rate goes down with increasing Mach number. Thus, the change in moment about the center of gravity is decreased, and the damping from the tail is less effective.

#### 4.5 Roll Damping Derivative

Prescribed motion roll tests were run for six Mach numbers from 1.58 to 2.50,  $\alpha = 0^\circ$  with the inviscid grids. As a reminder, the values chosen to define the motion and temporal discretization were  $k_p = 0.0025$  and 23,040 iterations per revolution. As a contrast to the dynamic pitch cases, the dynamic roll tests were run to convergence rather than freely or with periodic motion, eliminating the need for time-accurate solving. Recall that, from a converged solution, the roll damping coefficient is defined as  $C_{l_p} = \frac{C_l}{k_p}$ .

Figure 4.31 shows the  $C_l$  histories of the six Mach numbers tested. Convergence



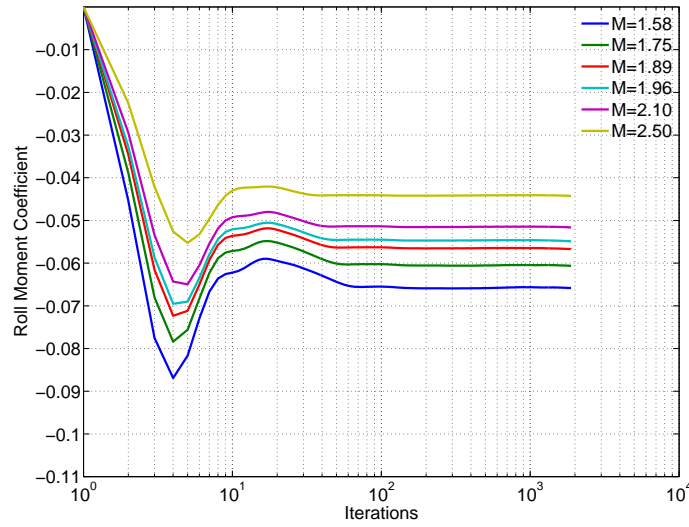


Figure 4.31: Roll moment convergence for six Mach numbers,  $k_p = 0.0025$ .

was achieved within approximately one degree of roll, but each case was run to a roll angle of  $45^\circ$  in an attempt to remove any variations in the measured value. Regardless of how far they were run, however, a certain amount of random noise was observed in the roll moment coefficient measurement. This noise was considered insignificant, however, because the value never strayed by more than 0.5% from the mean value for each case. This mean convergence value was used in calculations of  $C_{l_p}$ .

The converged value from each of the Mach numbers was used to determine the local roll damping coefficient. The magnitude of the roll damping was found to decrease steadily with increasing Mach number, as seen in Figure 4.32. This is due to the fact that, as Mach number increases, the induced velocity on the tail fins due to roll becomes a lesser percentage of the total velocity. This reduces the induced angle of attack, and thus the total moment produced by the tail. The tail continues to resist rolling motion at all Mach numbers, but to a lesser extent at faster speeds. Comparison data showed some degree of scatter for the roll damping of similar cases, but, in general, the present methods show very good agreement with both experimental and computational data.

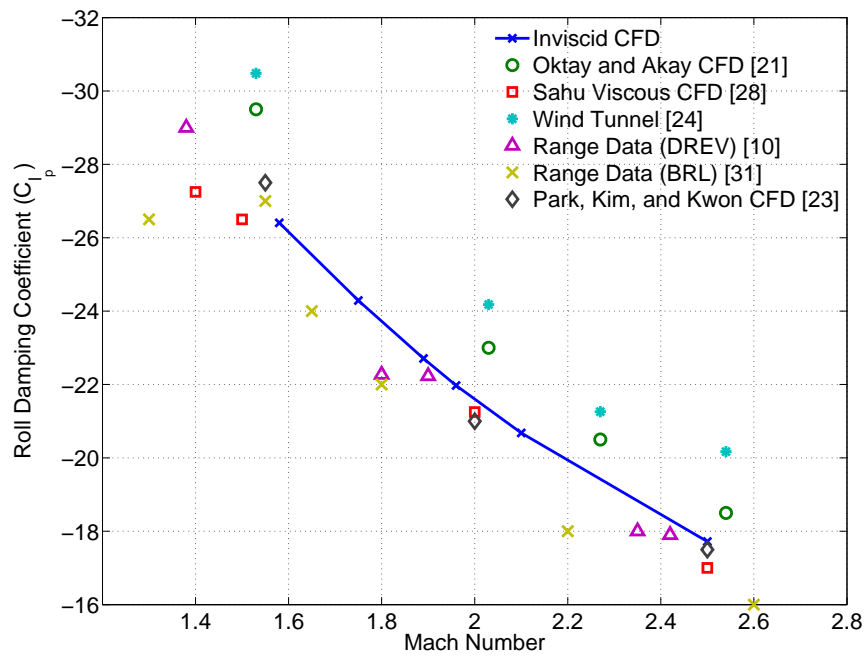


Figure 4.32: Roll damping coefficient as a function of Mach number,  $k_p = 0.0025$ .

## V. Conclusions

The use of the Beggar code for determining the static and dynamic pitch and roll stability derivatives of the Basic Finner missile was analyzed and verified. Various methods for determining the pitch derivatives were compared and shown to be in good agreement with one another and with experimental data. The static pitch stability derivatives were found from inviscid static solutions, inviscid forced oscillation, inviscid free oscillation, and viscous forced oscillations. The dynamic pitch derivatives were found from inviscid forced oscillation, inviscid free oscillation, and viscous forced oscillation. Dynamic roll derivatives were found from forced roll motion with constant angular rate. The parameters defining the forced pitch and roll motions were carefully chosen through multiple convergence studies. For pitch motion, these parameters were reduced pitch rate, amplitude, Newton iterations, iterations per oscillation, and total number of oscillations. Convergence studies were performed on reduced roll rate and iterations per revolution to define the rolling motion.

At all angles of attack and Mach number, the varying inviscid methods showed impressive consistency for determining the stability coefficients. However, only at angles of attack less than or equal to  $10^\circ$  did the predicted stability coefficients match well with wind tunnel and ballistic range data. Above  $\alpha = 10^\circ$ , the inviscid solver failed to model nonlinear separation effects, and this was found to degrade solution quality.

Viscous cases were found to agree well with the inviscid CFD, and not as well as expected with experimental data. Based on flow visualization, the viscous solver appears to have resolved the vortices caused by flow separation at high angles of attack, but the integration of forces and moments over the surface failed to bear out this difference between viscous and inviscid solutions. This unexpected result could be due to the use of a turbulence model not designed to model separation, a lack of full convergence for the viscous cases, insufficient grid resolution, or the fact that the sharp leading edges of the fins caused there to be little difference in surface pressure for the inviscid and viscous cases.

Aside from the issues with the viscous solutions, the methods used here showed that Beggar may quite readily be used to determine the stability derivatives of a supersonic projectile. The free oscillation method provides the most capability due to the fact that a single test may be used to determine the stability coefficients for a wide range of angle of attack. Forced oscillation techniques were shown to accurately compute local stability derivatives, but in order to find the derivatives as a function of angle of attack, multiple dynamic solutions were required. Each forced oscillation solution was less time-consuming than a free oscillation solution, but sufficiently resolving the static pitch stability and pitch damping as a function of angle of attack required multiple tests, which negated this advantage.

### ***5.1 Future Research***

Beggar has been shown to be a useful tool for determining fast estimates for the stability derivatives of a supersonic projectile, but full validation of this capability for a wide range of cases will require additional testing. This validation should be accomplished with additional models, methods, and flow regimes.

One validation case that would be particularly useful would be to perform tests similar to those performed here on the Army-Navy Spinner Rocket (ANSR). Like the Basic Finner, the stability coefficients of the ANSR are well documented in the literature, providing excellent sources of comparison. This case would also expand the application of current methods to a spin-stabilized projectile.

In addition to the methods shown here, steady state coning motion should be tested and verified. Such a steady state method is desirable because it eliminates the need for time-accurate solutions, which are typically difficult to run and computationally expensive.

The test methods performed here have been shown to work well with a supersonic projectile. Additional testing is required to validate the use of Beggar with these methods for subsonic, transonic, and higher supersonic Mach numbers than

were tested here. More complicated, asymmetric geometries should also be tested to verify the utility of these methods for more than the simple geometry tested here. Final testing with the Basic Finner could be expanded to include fin deflection to determine changes in the stability coefficients as a function of fin deflection angle.

Finally, once Beggar has been fully validated as an aerodynamic derivative prediction tool, it could be used to build a database of static stability coefficients. Such a database would allow missile designers to swiftly and accurately predict flight trajectories of new designs. This database and additional testing with Beggar could be used in early design phases to improve the final performance and to minimize the research and testing costs of new systems.

## Appendix A. Listings

### A.1 Beggar Inputs

Listing A.1: Beggar input file: finner.in. (Appendix2/finner.in)

```
#-----
#
#                               INITIALIZATION PARAMETERS
#-----
5 verbose = 3                                # Chapter 1, para. 4
  ptol = 1e-7                                # Chapter 9, para. 1
  nopatch                                    # Chapter 10, para. 4

10 cfl=250000
  init from '/home/scratch2/mbartowi/thesis/finner/static/d_m196/...
    a_a00/carraige.r01000'

  #dump plot3d every 40

15 #-----
#
#                               FLOW PROPERTIES
#                               (Ref. Beggar Manual, Chapter 4, para. 1)
#-----
20 #

  mach = 1.96

  # Rotate to desired angle of attack
25  rot z -0.0

#-----
#
#                               SIX+DOF PARAMETERS
#                               (Ref. Beggar Manual, Chapter 5, para. 4 & 5)
#-----
30 #

  sixdof gravity = <0.0,-32.18,0> # ft/s^2 for AOA=0 deg
35 sixdof density = 0.001262      # Slug/ft^3 @20000 ft
  sixdof soundspd = 1037.0        # ft/s @20000 ft
  sixdof refl = 0.104167          # Reference length conversion (ft)

#-----
40 #
#                               DYNAMIC CASE PRE-CHECK
#                               (Ref. Beggar Manual, Chapter 5, para. 17)
#-----
45 #noflow
```

```

#noflow_assembly

dt=0.034968731
50 #-----
#
#                               FLOW SOLVER PARAMETERS
#                               (Ref. Beggar Manual, Chapter 7)
55 #-----
#

stencil= inviscid2
solver = second order, full, euler, steg_warm_xair jacobians,
60     implicit bcupdate, primitive extrap, steger_warming ...
        right_side,
        three point backward time

65 dtiter = 4                # Newton Iterations [1]
   dtiter_tol = -10         # Newton iteration tolerance, [-10]
   inner = 80               # Gauss-Sidel Iterations [80]

#   BC update weighting
70   bcrelax = 1.0           #default = 1.0
   block to block relax = 0.4      #default = 0.4

#-----
#
75 #                               GRID ASSEMBLY
#-----
#
#                               Configure Basic Configuration to include
#                               Global Cartesian Grid
80 #                               Aircraft
#                               Pylons, Racks, and Launchers
#                               Pods, Tanks, and other stores
#                               Store of Interest
#
85 #-----
#                               Inertial Grid:  SB 1
#-----

readgrids '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/...
          geometry/inviscid/inert.p3ds' as plot3d ascii
90

#-----
#                               Store of Interest:  SB 2-6
#-----
95 # Here include the store, the fins, the fspec, and the dyn.

```

```

include '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/beg/...
inviscid/body.beg'

include '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/beg/...
inviscid/fin1.beg'
100 include '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/beg/...
inviscid/fin2.beg'
include '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/beg/...
inviscid/fin3.beg'
include '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/beg/...
inviscid/fin4.beg'

include '/home/scratch2/mbartowi/thesis/finner/dynamic/pitch/...
motion_files/mach_196.dyn'
105 include '/home/afiten1/gae08m/mbartowi/scratch/thesis/finner/beg/...
inviscid/finner.fspect'

#-----
#
#           SET WORLDSIDE FOR CCUT OPTIONS
110 #           (Ref. Beggar Manual, Chapter 8, para. 6)
#-----
#

# Set worldside cell
115 sb 2
    g 1
        set (10,30,4) (11,31,5) to worldside

```

Listing A.2: Beggar grid input file: fin2.beg. (Appendix2/fin2.beg)

```

directory prefix = '/home/afiten1/gae08m/mbartowi/scratch/thesis/...
finner/geometry/inviscid/'
readgrids 'fin.p3df' as plot3d ascii
tag 'fin2_SB'

5 #----- Set BCs -----
g 1
    set "fin2" = (59,1,1) (11,20,1) to tangent
    set (59,1,*) (11,1,1) to tangent nocut
10 g 2
    set "fin2" += (1,1,1) (*,*,1) to tangent
g 3
    set (1,1,1) (*,*,1) to tangent nocut
g 4
15 set "fin2" += (1,1,18) (*,1,1) to tangent

20 #----- Set protected cells -----
g 1

```



```

    set (11,1,1) (59,3,*) to protect
g 3
    set (1,1,1) (*,*,3) to protect
25

```

```

#----- Rotate -----
30 rot x 90

```

Listing A.3: Beggar force specification file: finner.fspect. (Appendix2/finner.fspect)

```

# -----
#
#                               FORCE SPECIFICATIONS
#                               (Ref. Beggar Manual, Chapter 8)
5 # -----
# -----TOTAL FSPEC-----
#
forcespec "finner": dump every 1 to "finner.forces"
10 with refl=1.0                # Approx diameter of store (grid ...
    units)
    with refa=0.7854            # Approx cross sectional area ...
    using body diameter (grid units)
    with mcenter = <6.1,0.0,0.0> # grid units
forcespec "finner": add "body"
forcespec "finner": add "fin1"
15 forcespec "finner": add "fin2"
forcespec "finner": add "fin3"
forcespec "finner": add "fin4"
# -----

```

Listing A.4: Beggar dynamic specification file: mach196.dyn. (Appendix2/mach196.dyn)

```

# -----
#
#                               INERTIAL AND DYNAMIC SPECIFICATIONS
#                               (Ref. Beggar Manual, Chapter 9)
5 # -----
# -----Body-----
#
10 dynamicspec "basic":
    add sb 'body_SB';
    add sb 'fin1_SB';
    add sb 'fin2_SB';
    add sb 'fin3_SB';
15 add sb 'fin4_SB';

```

```

##          INERTIAL PROPERTIES / FORCE SPECIFICATIONS
20      add fspec 'finner';
      cg = <0.63542,0.0,0.0>;          # ft 6.1 calibers
      spec_motion_file = "/home/scratch2/mbartowi/thesis/finner/...
          dynamic/pitch/motion_files/mach_196.dat";
      trelease = 0.0;

25

##          DYNAMIC DATA FILE CREATION

      read gandc z down gviz;
30      dump gandc z down

```

Listing A.5: Beggar prescribed motion file: mach196.dat. (Appendix2/mach196.dat)

```

* Generated using transform.m
* Angular rate = 9.8 rad/s
* Reduced frequency = 0.0003
* Magnitude of oscillation = 0.5 degrees
5 * Number of complete oscillations = 3.0
* Dimensionless Timestep = 0.03496873
* Physical Timestep = 3.51262018e-06
* Total time = 0.01826562 seconds
* Iterations = 5201
10 * Euler rotation order about CFD axis [1 3 2]
* dt          dx dy dz th_x theta_y theta_z
0              0 0 0 0 0 -0
3.512620184e-06 0 0 0 0 0 -0.001963490362
7.025240368e-06 0 0 0 0 0 -0.003926950444
15 1.053786055e-05 0 0 0 0 0 -0.005890349968
1.405048074e-05 0 0 0 0 0 -0.007853658656
1.756310092e-05 0 0 0 0 0 -0.00981684623
2.10757211e-05 0 0 0 0 0 -0.01177988242
2.458834129e-05 0 0 0 0 0 -0.01374273694
20 2.810096147e-05 0 0 0 0 0 -0.01570537954

      .          . . . . .
      .          . . . . .
      .          . . . . .

25      0.001390997593 0 0 0 0 0 -0.4999383162
      0.001394510213 0 0 0 0 0 -0.4999653026
      0.001398022833 0 0 0 0 0 -0.4999845788
      0.001401535453 0 0 0 0 0 -0.4999961447
30 0.001405048074 0 0 0 0 0 -0.5
      0.001408560694 0 0 0 0 0 -0.4999961447
      0.001412073314 0 0 0 0 0 -0.4999845788
      0.001415585934 0 0 0 0 0 -0.4999653026
      0.001419098554 0 0 0 0 0 -0.4999383162
35      .          . . . . .

```

```

      .      .      .      .      .      .
      .      .      .      .      .      .
40 0.00420109374    0    0    0    0    0    0.4999383162
    0.00420460636    0    0    0    0    0    0.4999653026
    0.00420811898    0    0    0    0    0    0.4999845788
    0.004211631601    0    0    0    0    0    0.4999961447
    0.004215144221    0    0    0    0    0    0.5
45 0.004218656841    0    0    0    0    0    0.4999961447
    0.004222169461    0    0    0    0    0    0.4999845788
    0.004225682081    0    0    0    0    0    0.4999653026
    0.004229194701    0    0    0    0    0    0.4999383162

```

Listing A.6: Beggar execution file: rubeg. (Appendix2/runbeg)

```

#PBS -l nodes=4:ppn=2
#PBS -j oe
#PBS -M michael.bartowitz@afit.edu
#PBS -N angle_00
5
MPICHBIN=/apps/Linux86_64/partools/mpich-1.2.7p1/bin
Beg=/apps/ECS/Beggar/Beg117j/opteron.mpich/dp/opt/Beg.mpich

nprocs=$(cat $PBS_NODEFILE|wc -l)
10 nnodes=$(cat $PBS_NODEFILE|sort -u|wc -l)
echo $nprocs
echo $nnodes

cd $PBS_O_WORKDIR
15
# now run beggar

$MPICHBIN/mpirun -machinefile $PBS_NODEFILE -np $nprocs $Beg -hgl...
    =15 -dcut=2 -Direset -motion -i 5200 -r finner.r00000 > r650....
    out

```

## A.2 Post-processing Tools

Listing A.7: Matlab® post-processing tool. (Appendix2/postprocess.m)

```

% -----...
%
%           Read in and plot Free Oscillation Data
% -----...

5 name = 'finner.forces';
  angle_start = #;

data = dlmread(name, '', 120, 0);

```

```

10 time = data(:,2);

    % local aero force and moment coefficients
    cf_loc_aero_x = data(:,3);
    cf_loc_aero_y = data(:,4);
15 cf_loc_aero_z = data(:,5);

    cm_loc_aero_x = data(:,6);
    cm_loc_aero_y = data(:,7);
    cm_loc_aero_z = data(:,8);
20
    % position and orientation
    dx = data(:,9);
    dy = data(:,10);
    dz = data(:,11);
25
    th_x = data(:,12);
    th_y = data(:,13);
    th_z = data(:,14);

30 % global aero force and moment coefficients
    cf_glb_x = data(:,15);
    cf_glb_y = data(:,16);
    cf_glb_z = data(:,17);

35 cm_glb_x = data(:,18);
    cm_glb_y = data(:,19);
    cm_glb_z = data(:,20);

    % local total forces and moments
40 f_x = data(:,21);
    f_y = data(:,22);
    f_z = data(:,23);

    m_x = data(:,24);
45 m_y = data(:,25);
    m_z = data(:,26);

    cf_y = cf_loc_aero_y;
    cm_z = -cm_loc_aero_z;
50 th_z = -th_z+angle_start;

    L = length(time);

    % -----
55 %           Pitch Derivatives
    % -----

    % ***** Forced Oscillation Cases *****
    % Locations of interest, assuming 3.25 oscillations
60 l = floor(11/13*L);      % left, corresponds to minimum angle
    r = floor(9/13*L);      % right, corresponds to maximum angle

```

```

t = floor(10/13*L);      % top, corresponds to pitching down
b = floor(12/13*L);      % bottom, corresponds to pitching up

65 th_lt = th_z(l);
   th_rt = th_z(r);

   cm_lt = cm_z(l);
   cm_rt = cm_z(r);
70
   cm_up = cm_z(b);
   cm_dn = cm_z(t);

75 % Static Stability Derivative
   stab_stat = ( cm_rt-cm_lt ) ./ ( th_rt-th_lt );      % per deg
   stab_stat = stab_stat * (180/pi);                  % per rad

   % Dynamic Stability Derivative
80 stab_dyn = ( cm_up-cm_dn ) ./ (2*k);              % per reduced freq

   % ***** Free Oscillation Cases *****
   %find indices of angle of interest
85 alpha=#;      % Choose # to search for
   j = 1;
   k = 1;
   for i = 2:length(th_z)
       if k*(th_z(i)-alpha) < 0 % enter if statement after a ...
           crossing
90         ind(j) = i;
           j=j+1;
           k=k*-1;
       end
   end
95
   % interpolate to find coefficient at angle of interest
   cm_slope = (cm_z(ind)-cm_z(ind-1)) ./ (th_z(ind)-th_z(ind-1));
   cm = cm_z(ind) + cm_slope .* ( alpha-th_z(ind) );

100 q = deg2rad(( th_z(ind)-th_z(ind-1) )) ./ ( dt_phys );

   k = (q*d) ./ (2*V);

   P = polyfit(k,cm,1);
105 damping = P(1);
   cm_static = P(2); % use this with consecutive angles to get ...
       stiffness

% -----
110 % Roll Derivatives
% -----

```

```
% Dynamic Stability Derivative  
damp = cm_x_conv ./ k;    % uses converged value of cm_x
```

## Bibliography

1. “Viscous Grid Spacing Calculator”. <http://geolab.larc.nasa.gov/APPS/YPlus/>, January 2008.
2. Air Force SEEK Eagle Office. *Beggar Version 110C+ User’s Manual*, March 2006.
3. Anderson, John D. *Modern compressible Flow*. McGraw Hill, third edition, 2003.
4. Blazek, J. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, second edition, 2005.
5. Boissevain, Alfred G. and Peter F. Intrieri. *Determination of Stability Derivatives from Ballistic Range Tests of Rolling Aircrafts Models*. Technical report, National Aeronautics and Space Administration, January 1961.
6. Boyce, William E. and Richard C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. Wiley, 7th edition, 2000.
7. Chapman, Gary T. and Leslie A. Yates. “Aerodynamic Flight Simulation and CFD”. Presented at AFRL/VA Wright-Patterson AFB, May 2007.
8. DeSpirito, James, Sidra I. Siltan, and Paul Weinacht. “Navier-Stokes Predictions of Dynamic Stability Derivatives: Evaluation of Steady-State Methods”. *40th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2008-214. January 2008.
9. Devan, L., L. A. Mason, and F. G. Moore. *Aerodynamics of Tactical Weapons to Mach Number 8 and Angle-of-Attack of 180 degrees*. Technical Report ADA111765, Naval Surface Weapons Center, May 1981.
10. Devan, Leroy. *Nonaxisymmetric Body, Supersonic, Inviscid Dynamic Derivative Prediction*. Technical Report 89-2195-CP, Naval Surface Warfare Center, June 1989.
11. Fleeman, Eugene L. *Tactical Missile Design*. American Institute of Aeronautics and Astronautics, second edition, 2006.
12. James M. Brock, Jr and Bruce Jolly. “Application of Computational Fluid Dynamics at Eglin Air Force Base”. AIAA, 1998.
13. Karnopp, Dean. *Vehicle Stability*. Marcell Dekker, 2004.
14. Lee, Jae M., Kevin S. Dunworth, Magdi Rizk, William S. Westmoreland, and Donald J. Atkins. “Studies of Combined use of CFD and Wind Tunnel Test Approaches to Simulate a Store Separation from F-15E Using Efficient CFD Database Generation”. *22nd Applied Aerodynamics Conference and Exhibit*, AIAA 2004-4724. August 2004.
15. Maple, Raymond C. “AERO 652 Computational Fluid Dynamics”, 2007. Class Notes.

16. Murman, Scott M. "A Reduced-Frequency Approach for Calculating Dynamic Derivatives". *43rd AIAA Aerospace Sciences Meeting*, AIAA 2005-0840. January 2005.
17. Murman, Scott M. and Michael J. Aftosmis. "Dynamic Analysis of Atmospheric-Entry Probes and Capsules". *45th AIAA Aerospace Sciences Meeting*, AIAA 2007-0074. January 2007.
18. Nelson, Robert C. *Flight Stability and Automatic Control*. McGraw Hill, second edition, 1998.
19. Noack, Ralph W. and Bruce Jolly. "Fully Time Accurate CFD Simulations of JDAM Separation from an F-18C Aircraft". *38th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2000-0794. January 2000.
20. Ogata, Katsuhiko. *Modern Control Engineering*. Prentice-Hall, fourth edition, 2002.
21. Oktay, Erdal and Hasan U. Akay. "CFD Predictions of Dynamic Derivatives for Missiles". *40th AIAA Aerospace Sciences Meeting & Exhibit*, AIAA 2002-0276. January 2002.
22. Orlik-Ruckemann, K. J. "Dynamic Stability Testing of Aircraft - Needs Versus Capabilities". *Prog. Aerospace Sci.*, 16:431-447, 1975.
23. Park, Soo Hyung, Yoonsik Kim, and Jang Hyuk Kwon. "Prediction of Dynamic Damping Coefficients Using Unsteady Dual-Time Stepping Method". *40th AIAA Aerospace Sciences Meeting & Exhibit*, AIAA 2002-0715. January 2002.
24. Regan, F. J. *Roll Damping Moment Measurements for the Basic Finner at Subsonic and Supersonic Speeds*. NAVORD Report 6652, U.S. Naval Ordnance Laboratory, March 1964.
25. Rizk, Magdi, Steven Ellison, and Nathan C. Prewitt. "Beggar - A Store Separation Predictive Tool". *32nd AIAA Fluid Dynamics Conference and Exhibit*, AIAA 2002-3190. June 2002.
26. Rizk, Magdi and Jae M. Lee. "Beggar Code Implementation of the (6+)DOF Capability for Stores with Moving Components". *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2004-1251. January 2004.
27. Sahu, Jubaraj. "Time-Accurate Numerical Prediction of Free Flight Aerodynamics of a Finned Projectile". *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA 2005-3817. August 2005.
28. Sahu, Jubaraj. "Numerical Computations of Dynamic Derivatives of a Finned Projectile using a Time-Accurate CFD Method". *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA 2007-6581. August 2007.
29. Sahu, Jubaraj. "Unsteady Flow Computations of a Finned Body in Supersonic Flight". *25th AIAA Applied Aerodynamics Conference*, AIAA 2007-3933. June 2007.



30. Sahu, Jubaraj. "Time-Accurate Computations of Dynamic Derivatives of Projectiles". *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2008-0213. January 2008.
31. Shantz, Irving and Robert T. Groves. *Dynamic and Static Stability Measurements of the Basic Finner at Supersonic Speeds*. Technical Report NAVORD Report 4516, U.S. Naval Ordnance Laboratory, September 1960.
32. Stevens, Brian L. and Frank L. Lewis. *Aircraft Control and Simulation*. John Wiley and Sons, second edition, 2003.
33. Uselton, Bob L. and James C. Uselton. *Test Mechanism for Measuring Pitch-Damping Derivatives of Missile Configurations at High Angles of Attack*. Technical Report AD-A009 865, Arnold Engineering Development Center, May 1975.
34. Weinacht, Paul. "Prediction of projectile performance, stability and free-flight motion using CFD". *35th AIAA Aerospace Sciences Meeting & Exhibit*, AIAA 97-0421. January 1997.
35. Wheeler, Anthony J. and Ahmad R. Ganji. *Introduction to Engineering Experimentation*. Pearson Prentice-Hall, second edition, 2004.
36. White, Frank M. *Viscous Fluid Flow*. McGraw Hill, third edition, 2006.
37. Whitfield, David L. and Lafe K. Taylor. "Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes". *10th AIAA Computational Fluid Dynamics Conference*, AIAA 91-1539. June 1991.
38. Winchenbach, G. L. *Aerodynamic Testing in a Free-Flight Spark Range*. Technical Report WL-TR-1997-7006, Wright Laboratory Armament Directorate, April 1997.

<b>REPORT DOCUMENTATION PAGE</b>			Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>				
1. REPORT DATE (DD-MM-YYYY) 01-03-2008		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) Mar 2007 — Mar 2008
4. TITLE AND SUBTITLE  Determination of Static and Dynamic Stability Derivatives Using Beggar			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)  Bartowitz, Michael E., 2 Lt, USAF			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/GAE/ENY/08-M02	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Air Force SEEK EAGLE Office ATTN: Joseph Keen, 46 SK/SKI 205 W D Blvd, Bldg 350, Ste 348 Eglin AFB FL 32542 DSN: 872-0932			10. SPONSOR/MONITOR'S ACRONYM(S) N/A	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A	
12. DISTRIBUTION / AVAILABILITY STATEMENT  APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT <p>The static and dynamic pitch and roll stability derivatives of a finned, axisymmetric missile known as the Basic Finner were examined using a Computational Fluid Dynamics (CFD) approach. Stability derivatives are used to characterize vehicle motion, and knowledge of them is critical to the design of stable uncontrolled vehicles and control systems for controlled vehicles. Using CFD to characterize the motion of new munition designs has the potential to improve overall performance and reduce research and testing costs. The present analysis simulated forced oscillation and free oscillation of the Basic Finner model using the Air Force SEEK EAGLE Office's Beggar code. The pitch stability derivatives were determined at 0 degrees angle of attack for six Mach numbers from 1.58 to 2.50 and at Mach number equal to 1.96 for angles of attack from 0 to 20 degrees. The parameters defining the motion of the forced oscillation tests were the reduced pitch rate, amplitude, Newton iterations, iterations per oscillation, and total oscillations. Convergence studies on each of these parameters were performed to ensure both convergence and solution independence. Roll stability derivatives were determined through forced, constant rate rolling motion for six Mach numbers from 1.58 to 2.50 at an angle of attack of 0 degrees. The parameters defining the roll motion were reduced roll rate and iterations per revolution, which were chosen in the same manner as the pitch parameters. Good agreement was found between the different methods tested, previous CFD analysis, and experimental data.</p>				
15. SUBJECT TERMS Static Stability, Dynamic Stability, Computational Fluid Dynamics, Beggar, Basic Finner, Fin Stabilized Ammunition, Missile Trajectory				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UU	18. NUMBER OF PAGES  122
a. REPORT  U	b. ABSTRACT  U	c. THIS PAGE  U		
			19a. NAME OF RESPONSIBLE PERSON Lt Col Raymond Maple, AFIT/ENY	
			19b. TELEPHONE NUMBER (Include Area Code) (937) 255-3636, ext 4577; e-mail: raymond.maple@afit.edu	